2015 IEEE 56th Annual Symposium on Foundations of Computer Science

A Robust Sparse Fourier Transform in the Continuous Setting

Eric Price
Department of Computer Science
The University of Texas at Austin
Austin, USA
ecprice@cs.utexas.edu

Zhao Song
Department of Computer Science
The University of Texas at Austin
Austin, USA
zhaos@utexas.edu

Abstract

In recent years, a number of works have studied methods for computing the Fourier transform in sublinear time if the output is sparse. Most of these have focused on the discrete setting, even though in many applications the input signal is continuous and naive discretization significantly worsens the sparsity level.

We present an algorithm for robustly computing sparse Fourier transforms in the continuous setting. Let $x(t) = x^*(t)+g(t)$, where x^* has a k-sparse Fourier transform and g is an arbitrary noise term. Given sample access to $x(t)$ for some duration T, we show how to find a k-Fourier-sparse reconstruction $x'(t)$ with

$$\frac{1}{T}\int_0^T |x'(t) - x(t)|^2 \mathrm{d}t \lesssim \frac{1}{T}\int_0^T |g(t)|^2 \mathrm{d}t.$$

The sample complexity is linear in k and logarithmic in the signal-to-noise ratio and the frequency resolution. Previous results with similar sample complexities could not tolerate an infinitesimal amount of i.i.d. Gaussian noise, and even algorithms with higher sample complexities increased the noise by a polynomial factor. We also give new results for how precisely the individual frequencies of x^* can be recovered.

Keywords

Fourier Transform; Sparse Recovery; Superresolution;

I. Introduction

The Fourier transform is ubiquitous in digital signal processing of a diverse set of signals, including sound, image, and video. Much of this is enabled by the Fast Fourier Transform (FFT) [1], which computes the n-point discrete Fourier transform in $O(n \log n)$ time. But can we do better?

In many situations, much of the reason for using Fourier transforms is because the transformed signal is *sparse*—i.e., the energy is concentrated in a small set of k locations. In such situations, one could hope for a dependency that depends nearly linearly on k rather than n. Moreover, one may be able to find these frequencies while only sampling the signal at a portion of time. This idea has led to a number of results on *sparse Fourier transforms*, including [2], [3], [4], [5], that can achieve $O(k \log(n/k) \log n)$ running time and $O(k \log(n/k))$ sample complexity (although not quite both at the same time) in a robust setting.

These works apply to the discrete Fourier transform, but lots of signals including audio or radio originally come from a continuous domain. The standard way to convert a continuous Fourier transform into a discrete one is to apply a window function then subsample. Unfortunately, doing so "smears out" the frequencies, blowing up the sparsity. Thus, one can hope for significant efficiency gains by directly solving the sparse Fourier transform problem in the continuous setting. This has led researchers to adapt techniques from the discrete setting to the continuous both in theory [6], [7] and in practice [8]. However, these results are not robust to noise: if the signal is sampled with a tiny amount of Gaussian noise or decays very slightly over time, no method has been known for computing a sparse Fourier transform in the continuous setting with sample complexity linear in k and logarithmic in other factors. That is what we present in this paper.

Formally, a vector $x^*(t)$ has a k-sparse Fourier transform if it can be written as

$$x^*(t) = \sum_{i=1}^{k} v_i e^{2\pi \mathbf{i} f_i t}$$

for some *tones* $\{(v_i, f_i)\}$. We consider the problem where we can sample some signal

$$x(t) = x^*(t) + g(t)$$

0272-5428/15 $31.00 © 2015 IEEE
DOI 10.1109/FOCS.2015.42

at any t we choose in some interval $[0, T]$, where $x^*(t)$ has a k-sparse Fourier transform and $g(t)$ is arbitrary noise. As long as g is "small enough," one would like to recover a good approximation to x (or to x^*, or to $\{(v_i, f_i)\}$) using relatively few samples $t \in [0, T]$ and fast running time. Our algorithm achieves several results of this form, but a simple one is an ℓ_2/ℓ_2 guarantee: we reconstruct an $x'(t)$ with k-sparse Fourier transform such that

$$\frac{1}{T}\int_0^T |x'(t) - x(t)|^2 \mathrm{d}t \lesssim \frac{1}{T}\int_0^T |g(t)|^2 \mathrm{d}t$$

using a number of samples that is k times logarithmic factors[1]. To the best of our knowledge, this is the first algorithm achieving such a constant factor approximation with a sample complexity sublinear in T and the signal-to-noise ratio.

Our algorithm also gives fairly precise estimates of the individual tones (v_i, f_i) of the signal x^*. To demonstrate what factors are important, it is helpful to think about a concrete setting. Let us consider sound from a simplified model of a piano.

Thought experiment: piano tuning: In a simplified model of a piano, we have keys corresponding to frequencies over some range $[-F, F]$. The noise $g(t)$ comes from ambient noise and the signals not being pure tones (because, for example, the notes might decay slowly over time). For concrete numbers, a modern piano has 88 keys spaced from about 27.5 Hz to $F = 4200$Hz. The space between keys ranges from a few Hz to a few hundred Hz, but most chords will have an $\eta = 30$Hz or more gap between the frequencies being played. One typically would like to tune the keys to within about $\pm\nu = 1$Hz. And piano music typically has k around 5.

Now, suppose you would like to build a piano tuner that can listen to a chord and tell you what notes are played and how they are tuned. For such a system, how long must we wait for the tuner to identify the frequencies? How many samples must the tuner take? And how robust is it to the noise?

If you have a constant signal-to-noise ratio, you need to sample for a time T of at least order $1/\nu = 1$ second in order to get 1Hz precision—frequencies within 1Hz of each other will behave very similarly over small fractions of a second, which noise can make indistinguishable. You also need at least $\Omega(k \log \frac{F}{k\nu}) \approx 50$ samples, because the support of the signal contains that many bits of information and you only get a constant number per measurement (at constant SNR). At higher signal-to-noise ratios ρ, these results extend to $\Omega(\frac{1}{\nu\rho})$ duration and $\Omega(k \log_\rho \frac{F}{k\nu})$ samples. But as the signal-to-noise ratio gets very high, there is another constraint on the duration: for $T < \frac{1}{\eta} \approx 33$ milliseconds the different frequencies start becoming hard to distinguish, which causes the robustness to degrade exponentially in k [9] (though the lower bound there only directly applies to a somewhat restricted version of our setting).

This suggests the form of a result: with a duration $T > \frac{1}{\eta}$, one can hope to recover the frequencies to within $\frac{1}{\rho T}$ using $O(k \log_\rho \frac{FT}{k})$ samples. We give an algorithm that is within logarithmic factors of this ideal: with a duration $T > \frac{O(\log(k/\delta))}{\eta}$, we recover the frequencies to within $O(\frac{1}{\rho T})$ using $O(k \log_\rho(FT) \cdot \log(k/\delta) \log k)$ samples, where ρ and $1/\delta$ are (roughly speaking) the minimum and maximum signal-to-noise ratios that you can tolerate, respectively.

Instead of trying to tune the piano by recovering the frequencies precisely, one may simply wish to record the sound for future playback with relatively few samples. Our algorithm works for this as well: the combination $x'(t)$ of our recovered frequencies satisfies

$$\frac{1}{T}\int_0^T |x'(t) - x(t)|^2 \mathrm{d}t \lesssim \frac{1}{T}\int_0^T |g(t)|^2 \mathrm{d}t.$$

Let us now state our main theorems. The first shows how well we can estimate the frequencies f_i and their weights v_i; we refer to this (v_i, f_i) pair as a *tone*.

Theorem I.1 (Tone estimation)**.** *Consider any signal $x(t) : [0, T] \to \mathbb{C}$ of the form*

$$x(t) = x^*(t) + g(t),$$

for arbitrary "noise" $g(t)$ and an exactly k-sparse $x^* = \sum_{i\in[k]} v_i e^{2\pi \mathbf{i} f_i t}$ with frequencies $f_i \in [-F, F]$ and frequency separation $\eta = \min_{i\neq j} |f_i - f_j|$. For some parameter $\delta > 0$, define the "noise level"

$$\mathcal{N}^2 := \frac{1}{T}\int_0^T |g(t)|^2 \mathrm{d}t + \delta \sum_{i=1}^k |v_i|^2.$$

[1]We use $f \lesssim g$ to denote that $f \leq Cg$ for some universal constant C.

We give an algorithm that takes samples from $x(t)$ over any duration $T > O(\frac{\log(k/\delta)}{\eta})$ and returns a set of k tones $\{(v_i', f_i')\}$ that approximates x^* with error proportional to $\mathcal{N}$. In particular, every large tone is recovered: for any v_i with $|v_i| \gtrsim \mathcal{N}$, we have for an appropriate permutation of the indices that

$$|f_i' - f_i| \lesssim \frac{\mathcal{N}}{T|v_i|} \qquad \textit{and} \qquad |v_i' - v_i| \lesssim \mathcal{N}. \tag{1}$$

In fact, we satisfy a stronger guarantee that the total *error is bounded:*

$$\sum_{i=1}^{k} \frac{1}{T} \int_0^T |v_i' e^{2\pi \mathbf{i} f_i' t} - v_i e^{2\pi \mathbf{i} f_i t}|^2 \mathrm{d}t \lesssim \mathcal{N}^2. \tag{2}$$

The algorithm takes $O(k \log(FT) \log(\frac{k}{\delta}) \log(k))$ samples and running time, and succeeds with probability at least $1 - 1/k^c$ for an arbitrarily large constant c.

We then show that the above approximation of the individual tones is good enough to estimate the overall signal $x(t)$ to within constant factors:

Theorem I.2 (Signal estimation)**.** *In the same setting as Theorem I.1, if the duration is slightly longer at $T > O(\frac{\log(1/\delta) + \log^2 k}{\eta})$, the reconstructed signal $x'(t) = \sum_{i=1}^{k} v_i' e^{2\pi \mathbf{i} f_i' t}$ achieves a constant factor approximation to the complete signal x:*

$$\frac{1}{T} \int_0^T |x'(t) - x(t)|^2 \mathrm{d}t \lesssim \mathcal{N}^2. \tag{3}$$

The algorithm takes $O(k \log(FT) \log(\frac{k}{\delta}) \log(k))$ samples and running time, and succeeds with probability at least $1 - 1/k^c$ for an arbitrarily large constant c.

The above theorems give three different error guarantees, which are all in terms of a "noise level" $\mathcal{N}^2$ that is the variance of the noise $g(t)$ plus δ times the energy of the signal. The algorithm depends logarithmically on δ, so one should think of $\mathcal{N}^2$ as being the variance of the noise, e.g. σ^2 if samples have error $N(0, \sigma^2)$.

Error guarantees.: Our algorithm does a good job of estimating the signal, but how exactly should we quantify this? Because very few previous results have shown robust recovery in the continuous setting, there is no standard error measure to use. We therefore bound the error in three different ways: the maximum error in the estimation of any tone; the weighted total error in the estimation of all tones; and the difference between the reconstructed signal and the true signal over the sampled interval. The first measure has been studied before, while the other two are to the best of our knowledge new but useful to fully explain the robustness we achieve.

The error guarantee (1) says that we achieve good recovery of any tones with magnitude larger than $C\mathcal{N}$ for some constant C. Note that such a requirement is necessary: for tones with $|v_i| \leq \mathcal{N}$, one could have $g(t) = -v_i e^{2\pi \mathbf{i} f_i t}$, completely removing the tone (v_i, f_i) from the observed $x(t)$ and making it impossible to find. For the tones of sufficiently large magnitude, we find them to within $\frac{\mathcal{N}}{T|v_i|}$. This is always less than $1/T$, and converges to 0 as the noise level decreases. This is known as *superresolution*–one can achieve very high frequency resolution in sparse, nearly noiseless settings. Moreover, by Lemma III.15 our "superresolution" precision $|f_i' - f_i| \lesssim \frac{\mathcal{N}}{T|v_i|}$ is optimal.

While the guarantee of (1) is simple and optimal given its form, it is somewhat unsatisfying. It shows that the maximum error over all k tones is $\mathcal{N}$, while one can hope to bound the *total* error over all k tones by $\mathcal{N}$. This is precisely what Equation (2) does. The guarantee (1) is the precision necessary to recover the tone to within $O(\mathcal{N})$ average error in time, that is (1) is equivalent to

$$\frac{1}{T} \int_0^T |v_i' e^{2\pi \mathbf{i} f_i' t} - v_i e^{2\pi \mathbf{i} f_i t}|^2 \mathrm{d}t \lesssim \mathcal{N}^2 \qquad \forall i \in [k].$$

In (2), we show that this bound holds even if we sum the left hand side over all $i \in [k]$, so the *average* error is a factor k better than would be implied by (1). It also means that the total mass of all tones that are not recovered to within $\pm 1/T$ is $O(\mathcal{N})$, not just that every tone larger than $O(\mathcal{N})$ is recovered to within $\pm 1/T$.

The stronger bound (2) can be converted to the guarantee (3), which is analogous to the ℓ_2/ℓ_2 recovery guarantee standard in compressive sensing. It bounds the error of our estimate in terms of the input noise, on average over the sampled duration. The standard form of the ℓ_2/ℓ_2 guarantee in the discrete sparse Fourier transform setting [2], [3], [4], [10], [5] compares the energy in frequency domain rather than time domain. This cannot be achieved directly in the continuous setting, since frequencies infinitesimally close to each other are indistinguishable over a bounded time interval T. But if the signal is periodic with period T (the case where a discrete sparse Fourier transform applies directly), then (3) is equivalent to the standard guarantee by Parseval's identity. So (3) seems like the right generalization of ℓ_2/ℓ_2 recovery to our setting.

Other factors.: Our algorithm succeeds with high probability in k, which could of course be amplified by repetition (but increasing the sample complexity and running time). Our running time, at $O(k \log(FT) \log(k/\delta) \log k)$, is (after translating between the discrete and continuous setting) basically a $\log k$ factor larger than the fastest known algorithm for discrete sparse Fourier transforms ([4]). But since that result only succeeds with constant probability, and repetition would also increase that by a $\log k$ factor, our running time is actually equivalent to the fastest known results that succeed with high probability in k.

Our sample complexity is $O(\log(k/\delta) \log k)$ worse than optimal, which is known to be possible in the discrete setting ([5]). However, the techniques that [5] uses to avoid the $\log(k/\delta)$ factor seem hard to adapt to the continuous setting without losing some of the robustness/precision guarantees.

Another useful property of our method, not explicitly stated in the theorem, is that the sampling method is relatively simple: it chooses $O(\log(FT) \log k)$ different random arithmetic sequences of points, where each sequence has $O(k \log(k/\delta))$ points spread out over a constant fraction of the time domain T. Thus one could implement this in hardware using a relatively small number of samplers, each of which performs regular sampling at a rate of $\frac{k \log(k/\delta)}{T}$. This is in contrast to the Nyquist rate for non-sparse signals of $2F$ — in the piano example, each sampler has a rate on the order of 50Hz rather than 8000Hz.

The superresolution frequency is optimal because two signals of magnitude v_i and frequency separation $\nu < 1/T$ will differ by $O(\nu^2 T^2 |v_i|^2)$ over the duration T, so for ν below our threshold the difference is just $\mathcal{N}^2$. Hence if the observed signal $x(t)$ looks identical to (v_i, f_i), it might actually be (v_i, f_i') with noise equaling the difference between the two.

The only previous result known in the form of (1) was [9], which lost a $\mathrm{poly}(k, \frac{\max |v_i|}{\min |v_i|}, \eta)$ factor in noise tolerance and also did not optimize for sample complexity.

A. Comparison to Naive Methods

This section compares our result to some naive ways one could try to solve this problem by applying algorithms not designed for a continuous, sparse Fourier transform setting. The next section will compare our result to algorithms that are designed for such a setting.

Nyquist Sampling.: The traditional theory of band-limited signals from discrete samples says that, from samples taken at the Nyquist rate $2F$, one can reconstruct an F-band-limited signal exactly. The Whittaker-Shannon interpolation formula then says that

$$x^*(t) = \sum_{i=-\infty}^{\infty} x^*(2Fi) \operatorname{sinc}(2Ft - i) \tag{4}$$

where $\operatorname{sinc}(t)$ is the normalized sinc function $\frac{\sin(\pi t)}{\pi t}$. This is for the band-limited "pure" signal x^*, but one could then get a relationship for samples of the actual signal $x(t)$. This has no direct implications for learning the tones (e.g. our (1) or (2)), but for learning the signal (our (3)) there is also an issue. Even in the absense of noise and for $k = 1$, this method will have error polynomially rather than exponentially small in the number of samples.

That is, if there is no noise the method has zero error given infinitely many samples. But we only receive samples over the interval $[0, T]$, leading to error. Consider the trivial setting of $x(t) = 1$. The partial sum of (4) at a given t will be missing terms for $i > 2Ft$ and $i < 0$, which (for a random t in $[0, T/2]$) have magnitude at most $1/(Ft)$. The terms alternate in sign, so the sum has error approximately $1/(Ft)^2$. This means that the error over the first $1/F$ time is a constant, leading to average error of $\frac{1}{FT}$. This is with an algorithm that uses FT samples and time. By contrast, our algorithm in the noiseless setting has error exponentially small in the samples and running time.

Discrete Sparse Fourier Transforms.: An option related to the previous would be to discretize very finely, then apply a discrete sparse Fourier transform algorithm to keep the sample complexity and runtime small. The trouble here is that sparse Fourier transforms require sparsity, and this process decreases the sparsity. In particular, this process supposes that the signal is periodic with period T, so one can analyze this process as first converting the signal to one, equivalent over $[0, T]$, but with frequency spectrum only containing integer multiples of $1/T$. This is done by convolving each frequency f_i with a sinc function (corresponding to windowing to $[0, T]$) then restricting to multiples of $1/T$ (corresponding to aliasing). The result is that a one-sparse signal $e^{2\pi \mathbf{i} f_i t}$ is viewed as having Fourier spectrum

$$\widehat{x''}[j] = \operatorname{sinc}(f_i T - j)$$

for $j \in \mathbb{Z}$. When f_i is not a multiple of $1/T$, this means the signal is not a perfectly sparse signal. And this is true regardless of the discretization level, which only affects the subsequent error from aliasing $j \in \mathbb{Z}$ down to $\mathbb{Z}_n$. To have error proportional to $\delta \|\widehat{x}^*\|_2$, one would need to run such methods for a sparsity level of k/δ. Thus, as with Nyquist sampling, the sample and runtime will be polynomial, rather than logarithmic, in δ.

The above discussion refers to methods for learning the signal (our (3)). In terms of learning the tones, one could run the algorithm for sparsity $O(k)$ so that δ is a small constant, which would let one learn roughly where the peaks are and get most of the frequencies to the nearest $1/T$. This would give a similar bound to our (1), but without the superresolution effect as the noise becomes small. On the plus side, the duration could be just $O(1/\eta)$—which is sufficient for the different peaks to be distinguishable—rather than $O(\frac{\log k}{\eta})$ as our method would require, and the time and sample complexities could save a $\log k$ factor (if one did not want to recover *all* the tones, just most of them).

Essentially, this algorithm tries to round each frequency to the nearest multiple of $1/T$, which introduces noise that is a constant fraction of the signal. If the signal-to-noise ratio is already low, this does not increase the noise level by that much so such an algorithm will work reasonably well. If the signal-to-noise ratio is fairly high, however, then the added noise leads to much worse performance. Getting a constant factor approximation to the whole signal is only nontrivial for high SNR, so such a method does badly in that setting. For approximating the tones, it is comparable to our method in the low SNR setting but does not improve much as SNR increases.

B. Previous Work In Similar Settings

There have been several works that recover continuous frequencies from samples in the time domain. Some of these are in our setting where the samples can be taken at arbitrary positions over $[0, T]$ and others are in the discrete-time (DTFT) setting where the samples must be taken at multiples of the Nyquist frequency $\frac{1}{2F}$.

The result of [7] shows that a convex program can solve the problem in the DTFT setting using $O(k \log k \log(FT))$ samples if the duration is $T > O(\frac{1}{\eta})$, in the setting where $g(t) = 0$ and the coefficients of x^* have random phases. The sample complexity is one log factor better than ours, which is what one would expect for the noiseless setting. They do not show robustness to noise, and the running time is polynomial in FT as well as k.

The result of [6] is in a similar setting to our paper, using techniques of the same lineage. It achieves very similar sample complexity and running time to our algorithm, and a guarantee similar in spirit to (1) with some notion of robustness. However, the robustness is weaker than ours in significant ways. They consider the noise $g(t)$ in frequency space (i.e. $\widehat{g}(f)$), then require that $\widehat{g}(f)$ is zero at any frequency within η of the signal frequencies f_i, and bound the error in terms of $\mathcal{N}' = \|\widehat{g}\|_1/k$ instead of $\|g\|_2$. This fails to cover simple examples of noise, including i.i.d. Gaussian noise $g(t) \sim N(0, \sigma^2)$ and the noise one would get from slow decay of the signal over time (e.g. $x(t) = x^*(t)e^{-\frac{t}{100T}}$.). Both types of noise violate both assumptions on the noise: $\widehat{g}(f)$ will be nonzero arbitrarily close to each f_i and $\|\widehat{g}\|_1$ will be unbounded. Their result also requires a longer duration than our algorithm and has worse precision for any fixed duration.

The result of [9] studies noise tolerance in the DTFT setting, ignoring sample complexity and running time. It shows that the matrix pencil method [11], using FT samples, achieves a guarantee of the form (1), except that the bounds are an additional $\mathrm{poly}(FT, k, \delta)$ factor larger. Furthermore, it shows a sharp characterization of the minimal T for which this is possible by any algorithm: $T = (1 \pm o(1))\frac{2}{\eta}$ is necessary and sufficient. It is an interesting question whether the lower bound generalizes to our non-DTFT setting, where the samples are not necessarily taken from an even grid.

Lastly, [8] tries to apply sparse Fourier transforms to a domain with continuous signals. They first apply a discrete sparse Fourier transform then use hill-climbing to optimize their solution into a decent set of continuous frequencies. They have interesting empirical results but no theoretical ones.

II. Algorithm Overview

At a high level, our algorithm is an adaptation of the framework used by [4] to the continuous setting. However, getting our result requires a number of subtle changes to the algorithm. This section will describe the most significant ones. We assume some familiarity with previous work in the area [2], [12], [3], [13], [4].

First we describe a high-level overview of the structure. The algorithm proceeds in $\log k$ stages, where each stage attempts to recover each tone with a large constant probability (e.g. 9/10). In each stage, we choose a parameter $\sigma \approx \frac{T}{k \log(k/\delta)}$ that we think of as "hashing" the frequencies into random positions. For this sigma, we will choose about $\log(FT)$ different random "start times" t_0 and sample an arithmetic sequence starting at t_0, i.e. observe

$$x(t_0), x(t_0 + \sigma), x(t_0 + 2\sigma), \ldots, x(t_0 + (k \log(k/\delta))\sigma)$$

We then scale these observations by a "window function," which has specific properties but among other things scales down the values near the ends of the sequence, giving a smoother transition between the time before and after we start/end sampling. We alias this down to $B = O(k)$ terms (i.e. add together terms $1, B + 1, 2B + 1, \ldots$ to get a B-dimensional vector) and take the B-dimensional DFT. This gives a set of B values $\widehat{u}_i$. The observation made in previous papers is that $\widehat{u}$ is effectively a *hashing* of the tones of $\widehat{x}$ into B buckets, where σ defines a permutation on the frequencies that affects

whether two different tones land in the same bucket, and $\widehat{u}_j$ approximately equals the sum of all the tones that land in bucket j, each scaled by a phase shift depending on t_0.

Because of this phase shift, for each choice of t_0 the value of $\widehat{u}_j$ is effectively a sample from the Fourier transform of a signal that contains only the tones of $\widehat{x}^*$ that land in bucket j, with zeros elsewhere. And since there are k tones and $O(k)$ buckets, most tones are alone in their bucket. Therefore this sampling strategy reduces the original problem of k-sparse recovery to one of 1-sparse recovery—we simply choose t_0 according to some strategy that lets us achieve 1-sparse recovery, and recover a tone for each bin.

One-sparse recovery.: The algorithm for one-sparse recovery in [4] is a good choice for adaptation to the continuous setting. It narrows down to the frequency in a locality-aware way, maintaining an interval of frequencies that decreases in size at each stage (in contrast to the method in [3], which starts from the least significant bit rather than most significant bit).

If a frequency is perturbed slightly in time (e.g., by multiplying by a very slow decay over time) this will blur the frequency slightly into a narrow band. The one-sparse recovery algorithm of [4] will proceed normally until it gets to the narrow scale, at which point it will behave semi-arbitrarily and return something near that band. This gives a desired level of robustness–the error in the recovered frequency will be proportional to the perturbation.

Still, to achieve our result we need a few changes to the one-sparse algorithm. One is related to duration: in the very last stage of the algorithm, when the interval is stretched at the maximal amount, we can only afford one "fold" rather than the typical $O(\log n)$. The only cost to this is in failure probability, and doing it for one stage is fine – but showing this requires a different proof. Another difference is that we need the final interval to have precision $\frac{1}{T\rho}$ if the signal-to-noise ratio is ρ—the previous analysis showed $\frac{1}{T\sqrt{\rho}}$ and needed to be told ρ, but (as we shall see) to achieve an ℓ_2/ℓ_2 guarantee we need the optimal ρ-dependence and for the algorithm to be oblivious to the value of ρ. Doing so requires a modification to the algorithm and slightly more clever analysis.

k-sparse recovery.: The changes to the k-sparse recovery structure are broader. First, to make the algorithm simpler we drop the [13]-style recursion with smaller k, and just repeat an $O(k)$-size hashing $O(\log k)$ times. This loses a $\log k$ factor in time and sample complexity, but because of the other changes it is not easy to avoid, and at the same time improves our success probability.

The most significant changes come because we can no longer measure the noise in frequency space or rely on the hash function to randomize the energy that collides with a given heavy hitter. Because we only look at a bounded time window T, Parseval's identity does not hold and the energy of the noise in frequency space may be unrelated to its observed energy. Moreover, if the noise consists of frequencies infinitesimally close to a true frequency, then because σ is bounded the true frequency will always hash to the same bin as the noise. These two issues are what drive the restrictions on noise in the previous work [6]—assuming the noise is bounded in ℓ_1 norm in frequency domain and is zero in a neighborhood of the true frequencies fixes both issues. But we want a guarantee in terms of the average ℓ_2 noise level $\mathcal{N}^2$ in time domain over the observed duration. If the noise level is $\mathcal{N}^2$, because we cannot hash the noise independently of the signal, we can only hope to guarantee reliable recovery of tones with magnitude larger than $\mathcal{N}^2$. This is in contrast to the $\mathcal{N}^2/k$ that is possible in the discrete setting, and would naively lose a factor of k in the ℓ_2/ℓ_2 approximation.

The insight here is that, even though the noise is not distributed randomly across bins, the total amount of noise is still bounded. If a heavy hitter of magnitude v^2 is not recovered due to noise, that requires $\Omega(v^2)$ noise mass in the bin that is not in any other bin. Thus the total amount of signal mass not recovered due to noise is $O(\mathcal{N}^2)$, which allows for ℓ_2/ℓ_2 recovery.

This difference is why our algorithm only gets a constant factor approximation rather than the $1+\epsilon$ guarantee that hashing techniques for sparse recovery can achieve in other settings. These techniques hash into $B = O(k/\epsilon)$ bins so the average noise per bin is $O(\frac{\epsilon}{k}\mathcal{N}^2)$. In our setting, where the noise is not hashed independently of the signal, this would give no benefit.

Another difference arises in the choice of the parameter σ, which is the separation between samples in the arithmetic sequence used for a single hashing, and gives the permutation during hashing. In the discrete setting, one chooses σ uniformly over n, which in our setting would correspond to a scale of $\sigma \approx \frac{1}{\eta}$. Since the arithmetic sequences have $O(k\log(k/\delta))$ samples, the duration would then become at least $\frac{k\log(k/\delta)}{\eta}$ (which is why [6] has this duration). What we observe is that σ can actually be chosen at the scale of $\frac{1}{k\eta}$, giving the desired $O(\frac{\log(k/\delta)}{\eta})$ duration. This causes frequencies at the minimum separation η to always land in bins that are a constant separation apart. This is sufficient because we use [4]-style window functions with strong isolation properties (and, in fact, [4] could have chosen $\sigma \approx n/B$); it would be an issue if we were using the window functions of [3], [5] that have smaller supports but less isolation.

Getting an ℓ_2 bound: Lastly, converting between the guarantee (2) to (3) is a nontrivial task that is trivial in the discrete setting. In the discrete setting, it follows immediately from the different frequencies being orthogonal to each other. In our setting, we use that the recovered frequencies should themselves have $\Omega(\eta)$ separation, and that well-separated frequencies are nearly orthogonal over long enough time scales $T \gg 1/\eta$. [2]

This bears some similarity to issues that arise in sparse recovery with overcomplete dictionaries. It would be interesting to see whether further connections can be made between the problems.

III. Proof outline

We do not have space to give the full proof, so in this section we present the key lemmas along the path to producing the algorithm.

Notation.: First we define the notation necessary to understand the lemmas. The full notation as used in the proofs appears in full version.

The algorithm proceeds in stages, each of which hashes the frequencies to B bins. The hash function depends on two parameters σ and b, and so we define it as $h_{\sigma,b}(f) : [-F, F] \to [B]$.

A tone with a given frequency f can have two "bad events" $E_{coll}(f)$ or $E_{off}(f)$ hold for a given hashing. These correspond to colliding with another frequency of x^* or landing within an α fraction of the edge, respectively; they each will occur with small constant probability.

For a given hashing, we will choose a number of different offsets a that let us perform recovery of the tones that have neither bad event in this stage.

Key Lemmas: First, we need to be able to compare the distance between two pure tone signals in time domain to their differences in parameters. The relation is as follows:

Lemma III.1. *Let (v, f) and (v', f') denote any two tones, i.e. (magnitude, frequency) pairs. Then for*

$$\operatorname{dist}\left((v,f),(v',f')\right)^2 := \frac{1}{T}\int_0^T \left|ve^{2\pi f t\mathbf{i}} - v'e^{2\pi f' t\mathbf{i}}\right|^2 \mathrm{d}t,$$

we have

$$\operatorname{dist}\left((v,f),(v',f')\right)^2 \eqsim (|v|^2 + |v'|^2)\cdot \min(1, T^2|f-f'|^2) + |v-v'|^2,$$

and

$$\operatorname{dist}\left((v,f),(v',f')\right) \eqsim |v|\cdot \min(1, T|f-f'|) + |v-v'|.$$

The basic building block for our algorithm is a function HashToBins, which is very similar to one of the same name in [4].

The key property of HashToBins is that, if neither "bad" event holds for a frequency f (i.e. it does not collide or land near the boundary of the bin), then for the bin $j = h_{\sigma,b}(f)$ we have that $|\widehat{u}_j| \approx |\widehat{x^*}(f)|$ with a phase depending on a.

How good is the approximation? In the discrete setting, one can show that each tone has error about $\mathcal{N}^2/B$ in expectation. Here, because the hash function cannot randomize the noise, we instead show that the total error over all tones is about $\mathcal{N}^2$:

Lemma III.2. *Let $\sigma \in [\frac{1}{B\eta}, \frac{2}{B\eta}]$ uniformly at random, then $b \in [0, \frac{\lceil F/\eta\rceil}{\sigma B}]$, $a \in [0, \frac{cT}{\sigma}]$ be sampled uniformly at random for some constant $c > 0$. Let the other parameters be arbitrary in $\widehat{u} = \mathsf{HashToBins}(x, P_{\sigma,a,b}, B, \delta, \alpha)$, and consider $H = \{f \in \operatorname{supp}(\widehat{x^*}) \mid$ neither $E_{coll}(f)$ nor $E_{off}(f)$ holds$\}$ and $I = [B] \setminus h_{\sigma,b}(\operatorname{supp}(\widehat{x^*}))$ to be the bins that have no frequencies hashed to them. Then*

$$\mathop{\mathbb{E}}_{\sigma,b,a}\left[\sum_{f\in H}\left|\widehat{u}_{h_{\sigma,b}(f)} - \widehat{x^*}(f)e^{a\sigma 2\pi f\mathbf{i}}\right|^2 + \sum_{j\in I}\widehat{u}_j^2\right] \lesssim \mathcal{N}^2$$

We prove Lemma III.2 by considering the cases of $x^* = 0$ and $g = 0$ separately; linearity then gives the result. Both follow from properties of our window functions.

Lemma III.3. *If $x^*(t) = 0, \forall t \in [0, T]$, then*

$$\mathop{\mathbb{E}}_{\sigma,a,b}[\sum_{j=1}^{B}|\widehat{u}_j|^2] \lesssim \frac{1}{T}\int_0^T |g(t)|^2 \mathrm{d}t.$$

[2] We use $f \gg g$ to denote that $f > Cg$ for any constant C.

Lemma III.4. *If $g(t) = 0, \forall t \in [0,T]$. Let H denote a set of frequencies, $H = \{f \in \text{supp}(\widehat{x}^*) \mid \text{neither } E_{coll}(f) \text{ nor } E_{off}(f) \text{ holds}\}$. Then,*

$$\mathop{\mathbb{E}}_{\sigma,a,b}[\sum_{f\in H}\left|\widehat{u}_{h_{\sigma,b}(f)} - \widehat{x^*}(f)e^{a\sigma 2\pi f\mathbf{i}}\right|^2] \le k\delta^2\|\widehat{x^*}\|_1^2.$$

Lemma III.2 is essentially what we need for 1-sparse recovery. We first show a lemma about the inner call, which narrows the frequency from a range of size Δl to one of size $\frac{\Delta l}{\rho st}$ for some parameters ρst. This gives improved performance (superresolution) when the signal-to-noise ratio ρ within the bucket is high. The parameter s and t provide a tradeoff between success probability, performance, running time, and duration.

Lemma III.5. *Consider any B, δ, α. Algorithm* HashToBins *takes $O(B\log(k/\delta))$ samples, runs in $O(\frac{B}{\alpha}\log(k/\delta) + B\log B)$ time.*

Lemma III.6. *Given σ and b, consider any frequency f for which neither $E_{coll}(f)$ nor $E_{off}(f)$ holds, and let $j = h_{\sigma,b}(f)$. Let $\mu^2(f) = \mathbb{E}_a[|\widehat{u}_j - \widehat{x^*}(f)e^{a\sigma 2\pi f\mathbf{i}}|^2]$ and $\rho^2 = |\widehat{x^*}(f)|^2/\mu^2(f)$. If $\rho > C$, (where C is known as the "approximation factor") then $\forall 0 < s < 1, t \ge 4$, consider any run of* LocateInner *with $f \in [l_j - \frac{\Delta l}{2}, l_j + \frac{\Delta l}{2}]$. It takes $O(R_{loc})$ random $(\gamma,\beta) \in [\frac{1}{2},1] \times [\frac{st}{4\sigma\Delta l}, \frac{st}{2\sigma\Delta l}]$ samples over duration $\beta\sigma = \Theta(\frac{st}{\Delta l})$, runs in $O(stR_{loc})$ time, to learn f within a region that has length $\frac{\Delta l}{t}$ with failure probability at most $(\frac{4}{s\rho})^{R_{loc}} + t\cdot(60s)^{R_{loc}/2}$.*

By repeating this inner loop, we can recover the tones in almost every bin that doesn't have the "bad" events happen, so we recover a large fraction of the heavy hitters in each stage.

Lemma III.7. *Algorithm* LocateKSignal *takes $O(k\log_C(FT)\log(k/\delta))$ samples over $O(\frac{\log(k/\delta)}{\eta})$ duration, runs in $O(k\log_C(FT)\log(FT/\delta))$ time, and outputs a set $L \subset [-F,F]$ of $O(k)$ frequencies with minimum separation $\Omega(\eta)$.*

Given σ and b, consider any frequency f for which neither of $E_{coll}(f)$ or $E_{off}(f)$ hold. Let $j = h_{\sigma,b}(f)$, $\mu^2(f) = \mathbb{E}_a[|\widehat{u}_j - \widehat{x^*}(f)e^{a\sigma 2\pi f\mathbf{i}}|^2]$, and $\rho^2 = |\widehat{x^*}(f)|^2/\mu^2(f)$. If $\rho > C$, then with an arbitrarily large constant probability there exists an $f' \in L$ with

$$|f - f'| \lesssim \frac{1}{T\rho}.$$

Combining this with estimation of the magnitudes of recovered frequencies, we can show that the total error over all bins without "bad" events—that is, bins with either one well placed frequency or zero frequencies—is small. At this point we give no guarantee for the (relatively few) bins with bad events; the recovered values there may be arbitrarily large.

Lemma III.8. *Algorithm* OneStage *takes $O(k\log_C(FT)\log(k/\delta))$ samples over $O(\frac{\log(k/\delta)}{\eta})$ duration, runs in $O(k(\log_C(FT)\log(FT/\delta)))$ time, and outputs a set of $\{(v_i', f_i')\}$ of size $O(k)$ with $\min_{i\neq j}|f_i' - f_j'| \gtrsim \eta$. Moreover, one can imagine a subset $S \subseteq [k]$ of "successful" recoveries, where $\Pr[i \in S] \ge \frac{9}{10}\ \forall i \in [k]$ and for which there exists an injective function $\pi : [k] \to [O(k)]$ so that*

$$\mathop{\mathbb{E}}_{\sigma,b}\left[\sum_{i\in S}\frac{1}{T}\int_0^T\left|v_i'e^{2\pi f_i' t\mathbf{i}} - v_{\pi(i)}e^{2\pi f_{\pi(i)}t\mathbf{i}}\right|^2\mathrm{d}t\right] \lesssim C^2\mathcal{N}^2.$$

with $1 - 1/k^c$ probability for an arbitrarily large constant c.

We can repeat the procedure for $O(\log k)$ stages and merge the results, getting a list of $O(k)$ tones that includes k tones that match up well to the true tones. However, we give no guarantee for the rest of the recovered tones at this point—as far as the analysis is concerned, mistakes from bins with collisions may cause arbitrarily large spurious tones.

Lemma III.9. *Repeating algorithm* OneStage *$O(\log k)$ times,* MergedStages *returns a set $\{(v_i', f_i')\}$ of size $O(k)$ with $\min_{i\neq j}|f_i' - f_j'| \gtrsim \eta$ that can be indexed by π such that*

$$\sum_{i=1}^k\frac{1}{T}\int_0^T\left|v_i'e^{2\pi f_i' t\mathbf{i}} - v_{\pi(i)}e^{2\pi f_{\pi(i)}t\mathbf{i}}\right|^2\mathrm{d}t \lesssim C^2\mathcal{N}^2.$$

with probability $1 - 1/k^c$ for an arbitrarily large constant c.

To address the issue of spurious tones, we run the above algorithm twice and only take the tones that are recovered in both stages. We show that the resulting $O(k)$ tones are together a good approximation to the vector.

Lemma III.10. *If we run* MergedStages *twice and take the tones* $\{(v_i', f_i')\}$ *from the first result that have* f_i' *within* $c\eta$ *for small* c *of some frequency in the second result, we get a set of* $k'' = O(k)$ *tones that can be indexed by some permutation* π *such that*

$$\sum_{i=1}^{k} \frac{1}{T} \int_0^T \left| v_i' e^{2\pi f_i' t \mathbf{i}} - v_{\pi(i)} e^{2\pi f_{\pi(i)} t \mathbf{i}} \right|^2 \mathrm{d}t + \sum_{i=k+1}^{k''} |v_i'|^2 \lesssim C^2 \mathcal{N}^2. \tag{5}$$

Simply picking out the largest k recovered tones then gives the result (2).

Theorem III.11. *Algorithm* ContinuousFourierSparseRecovery *returns a set* $\{(v_i', f_i')\}$ *of size* k *with* $\min_{i \neq j} |f_i' - f_j'| \gtrsim \eta$ *for which*

$$\sum_{i=1}^{k} \frac{1}{T} \int_0^T \left| v_i' e^{2\pi f_i' t \mathbf{i}} - v_{\pi(i)} e^{2\pi f_{\pi(i)} t \mathbf{i}} \right|^2 \mathrm{d}t \lesssim C^2 \mathcal{N}^2$$

with probability $1 - 1/k^c$ *for an arbitrarily large constant* c*.*

By only considering the term in the sum corresponding to tone i and applying Lemma III.1, we get result (1):

Corollary III.12. *With probability* $1 - 1/k^c$ *for an arbitrarily large constant* c*, we recover a set of tones* $\{(v_i', f_i')\}$ *such that, for any* v_i *with* $|v_i| \gtrsim \mathcal{N}$*, we have for an appropriate permutation of the indices that*

$$|f_i' - f_i| \lesssim \frac{\mathcal{N}}{T|v_i|} \qquad \text{and} \qquad |v_i' - v_i| \lesssim \mathcal{N}. \tag{6}$$

We then show that (2) implies (3) for sufficiently long durations T. A long duration helps because it decreases the correlation between η-separated frequencies.

Lemma III.13. *Let* $\{(v_i, f_i)\}$ *and* $\{(v_i', f_i')\}$ *be two sets of* k *tones for which* $\min_{i \neq j} |f_i - f_j| \geq \eta$ *and* $\min_{i \neq j} |f_i' - f_j'| \gtrsim \eta$ *for some* $\eta > 0$*. Suppose that* $T > O(\frac{\log^2 k}{\eta})$*. Then these sets can be indexed such that*

$$\frac{1}{T} \int_0^T |\sum_{i=1}^{k} (v_i' e^{2\pi \mathbf{i} f_i' t} - v_i e^{2\pi \mathbf{i} f_i t})|^2 \mathrm{d}t \lesssim \sum_{i=1}^{k} \frac{1}{T} \int_0^T |v_i' e^{2\pi \mathbf{i} f_i' t} - v_i e^{2\pi \mathbf{i} f_i t}|^2 \mathrm{d}t. \tag{7}$$

Combining Theorem III.11 and Lemma III.13 immediately implies

Theorem III.14. *Suppose we sample for a duration* $T > O(\frac{\log(1/\delta) + \log^2 k}{\eta})$*. Then the reconstructed signal* $x'(t) = \sum_{i=1}^{k} v_i' e^{2\pi \mathbf{i} f_i' t}$ *achieves a constant factor approximation to the complete signal* x*:*

$$\frac{1}{T} \int_0^T |x'(t) - x(t)|^2 \mathrm{d}t \lesssim C^2 \mathcal{N}^2 \tag{8}$$

The algorithm takes $O(k \log \frac{F}{\eta} \log(\frac{k}{\delta}) \log(k))$ *samples, runs in* $O(k \log \frac{F}{\eta} \log(\frac{k}{\delta}) \log(k))$ *time, and succeeds with probability at least* $1 - 1/k^c$ *for an arbitrarily large constant* c*.*

That finishes the proof of our main theorem. We also show that our "superresolution" precision from (1) is optimal, which is a simple corollary of Lemma III.1.

Lemma III.15. *There exists a constant* $c > 0$ *such that, for a given sample duration* T*, one cannot recover the frequency* f *to within*

$$c \frac{\mathcal{N}}{T|\widehat{x}^*(f)|}$$

with $3/4$ *probability, for all* $\delta > 0$*, even if* $k = 1$*.*

REFERENCES

[1] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[2] A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss, "Near-optimal sparse Fourier representations via sampling," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. ACM, 2002, pp. 152–161.

[3] A. C. Gilbert, S. Muthukrishnan, and M. Strauss, "Improved time bounds for near-optimal sparse Fourier representations," in *Optics & Photonics 2005*. International Society for Optics and Photonics, 2005, pp. 59 141A–59 141A.

[4] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, "Nearly optimal sparse Fourier transform," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM, 2012.

[5] P. Indyk and M. Kapralov, "Sample-optimal Fourier sampling in any constant dimension," in *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*. IEEE, 2014, pp. 514–523.

[6] P. Boufounos, V. Cevher, A. C. Gilbert, Y. Li, and M. J. Strauss, "What's the frequency, Kenneth?: Sublinear Fourier sampling off the grid," in *Algorithmica(A preliminary version of this paper appeared in the Proceedings of RANDOM/APPROX 2012, LNCS 7408, pp. 61-72)*. Springer, 2014, pp. 1–28.

[7] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, "Compressed sensing off the grid," *Information Theory, IEEE Transactions on*, vol. 59, no. 11, pp. 7465–7490, 2013.

[8] L. Shi, O. Andronesi, H. Hassanieh, B. Ghazi, D. Katabi, and E. Adalsteinsson, "Mrs sparse-fft: Reducing acquisition time and artifacts for in vivo 2d correlation spectroscopy," in *ISMRM13, Int. Society for Magnetic Resonance in Medicine Annual Meeting and Exhibition*, 2013.

[9] A. Moitra, "The threshold for super-resolution via extremal functions," in *STOC*, 2015.

[10] P. Indyk, M. Kapralov, and E. Price, "(Nearly) Sample-optimal sparse Fourier transform," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2014, pp. 480–499.

[11] Y. Hua and T. K. Sarkar, "Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38, no. 5, pp. 814–824, 1990.

[12] M. Charikar, K. Chen, and M. Farach-Colton, "Finding frequent items in data streams," in *Automata, Languages and Programming*. Springer, 2002, pp. 693–703.

[13] A. C. Gilbert, Y. Li, E. Porat, and M. J. Strauss, "Approximate sparse recovery: optimizing time and measurements," *SIAM Journal on Computing*, vol. 41, no. 2, pp. 436–453, 2012.

Due to limitations of space, we only provide the proofs for coverting (2) into (3), the lower bound proof for Lemma III.15, and pseudocode of the main algorithm in Algorithm 1 and 2. For the proofs of the remaining parts, see the full version.

IV. Proofs for converting (2) into (3)

In this section, we show that as long as the sample duration T is sufficiently large, it is possible to convert Equation (2) to Equation (3). First, we show an auxiliary lemma, Lemma IV.3, which bounds an integral that will appear in the analysis.

We will show that

$$\int_{-\infty}^{+\infty} \min(T, \frac{1}{|f_i-\theta|}) \cdot \min(T, \frac{1}{|f_j-\theta|}) \mathrm{d}\theta \lesssim \frac{\log(T|f_i-f_j|)}{|f_i-f_j|}.$$

for $f_j - f_i \geq 2/T$. We split this into two pieces.

Claim IV.1. *Given two frequencies f_i, f_j and $f_j - f_i \geq \frac{2}{T}$, we have*

$$\int_{f_i-\frac{1}{T}}^{f_i} \min(T, \frac{1}{|f_i-\theta|}) \cdot \frac{1}{|f_j-\theta|} \mathrm{d}\theta \lesssim \frac{1}{f_j-f_i}.$$

Proof: By $f_i - \frac{1}{T} < \theta < f_i$, we have

$$\mathrm{LHS} = \int_{f_i-\frac{1}{T}}^{f_i} T \cdot \frac{1}{f_j-\theta} \mathrm{d}\theta.$$

Since $\frac{1}{f_j-\theta} \eqsim \frac{1}{f_j-f_i}$ for all $\theta \in [f_i - \frac{1}{T}, f_i]$,

$$\mathrm{LHS} \lesssim \int_{f_i-\frac{1}{T}}^{f_i} \frac{T}{f_j-f_i} \mathrm{d}\theta = \frac{1}{f_j-f_i}.$$

■

Claim IV.2. *Given two frequencies f_i, f_j and $f_j - f_i \geq \frac{2}{T}$, we have*

$$\int_{-\infty}^{f_i-\frac{1}{T}} \min(T, \frac{1}{|f_i-\theta|}) \cdot \frac{1}{|f_j-\theta|} \mathrm{d}\theta \lesssim \frac{\log(T|f_j-f_i|)}{f_j-f_i}.$$

Proof: By $\theta < f_i - \frac{1}{T} < f_j$, we have that

$$\begin{aligned}
\mathrm{LHS} &= \int_{-\infty}^{f_i-\frac{1}{T}} \frac{1}{f_i-\theta} \cdot \frac{1}{f_j-\theta} \mathrm{d}\theta \\
&= \frac{1}{f_j-f_i} \int_{-\infty}^{f_i-\frac{1}{T}} \frac{f_j-f_i}{(f_i-\theta)(f_j-\theta)} \mathrm{d}\theta \\
&= \frac{1}{f_j-f_i} \int_{-\infty}^{f_i-\frac{1}{T}} \frac{1}{f_i-\theta} - \frac{1}{f_j-\theta} \mathrm{d}\theta \\
&= -\frac{1}{f_j-f_i} \log \frac{f_i-f_i+\frac{1}{T}}{f_j-f_i+\frac{1}{T}} \\
&= -\frac{1}{f_j-f_i} \log \frac{1}{T(f_j-f_i)+1} \\
&\lesssim \frac{\log(T(f_j-f_i))}{f_j-f_i}.
\end{aligned}$$

■

Lemma IV.3. *Given two frequencies f_i, f_j and $f_j - f_i \geq \frac{2}{T}$, we have*

$$\int_{-\infty}^{+\infty} \min(T, \frac{1}{|f_i-\theta|}) \cdot \min(T, \frac{1}{|f_j-\theta|}) \mathrm{d}\theta \lesssim \frac{\log(T|f_i-f_j|)}{|f_i-f_j|}.$$

Proof: By symmetry, we have

$$\mathrm{LHS} = 2 \int_{-\infty}^{\frac{f_i+f_j}{2}} \min(T, \frac{1}{|f_i-\theta|}) \cdot \min(T, \frac{1}{|f_j-\theta|}) \mathrm{d}\theta.$$

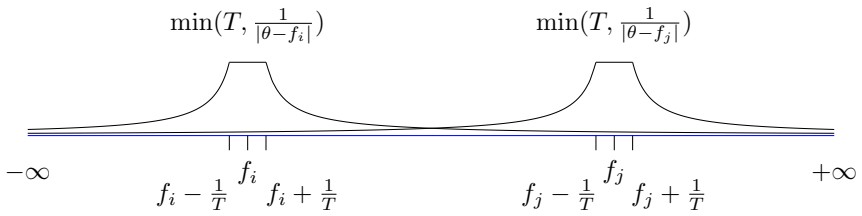

Figure 1: $\int_{-\infty}^{+\infty} \min(T, \frac{1}{|\theta-f_i|}) \cdot \min(T, \frac{1}{|\theta-f_j|}) \mathrm{d}\theta$

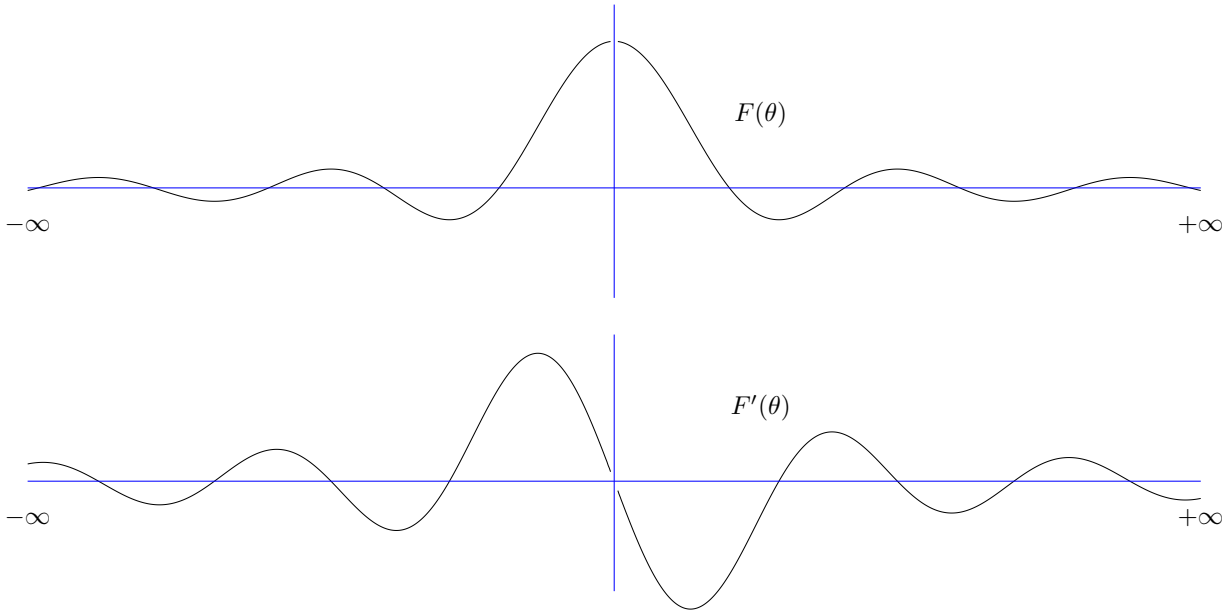

Figure 2: $F(\theta)$ is a sinc fuction and has derivate function $F'(\theta)$.

Since $T > \frac{1}{f_j-\theta}$ when $\theta < \frac{f_i+f_j}{2}$,

$$\text{LHS} \leq 2 \int_{-\infty}^{\frac{f_i+f_j}{2}} \min(T, \frac{1}{|f_i-\theta|}) \cdot \frac{1}{|f_j-\theta|} \mathrm{d}\theta.$$

We also observe that $\frac{1}{|f_j-\theta|} \eqsim \frac{1}{|f_j-f_i|}$ for all $\theta \in [f_i - \frac{f_j-f_i}{2}, f_i + \frac{f_j-f_i}{2}]$,

$$\int_{f_i-\frac{f_j-f_i}{2}}^{\frac{f_j+f_i}{2}} \min(T, \frac{1}{|f_i-\theta|}) \cdot \frac{1}{|f_j-\theta|} \mathrm{d}\theta \eqsim \int_{f_i-\frac{f_j-f_i}{2}}^{f_i} \min(T, \frac{1}{|f_i-\theta|}) \cdot \frac{1}{|f_j-\theta|} \mathrm{d}\theta.$$

Thus, we get

$$\text{LHS} \lesssim \int_{-\infty}^{f_i} \min(T, \frac{1}{|f_i-\theta|}) \cdot \frac{1}{|f_j-\theta|} \mathrm{d}\theta.$$

Plugging Claim IV.1 and IV.2 into the above formula completes the proof. ■

Lemma IV.4. *For any i, let $a_i(t) = v_i e^{2\pi f_i t \mathbf{i}} - v_i' e^{2\pi f_i' t \mathbf{i}}$, then for $i \neq j$,*

$$\frac{1}{T} \int_0^T a_i(t) \overline{a_j(t)} \mathrm{d}t \lesssim \frac{\log(\Delta f_{i,j} T)}{\Delta f_{i,j} T} \cdot \left(\frac{1}{T} \int_0^T |a_i(t)|^2 \mathrm{d}t \cdot \frac{1}{T} \int_0^T |a_j(t)|^2 \mathrm{d}t \right)^{\frac{1}{2}}, \tag{9}$$

where $\Delta f_{i,j} = \min(|f_i - f_j|, |f_i - f_j'|, |f_i' - f_j|, |f_i - f_j'|)$.

Proof: Let $\nu_i = |f_i - f_i'|$ and $\nu_j = |f_j - f_j'|$. Define $\|a_i\| = \sqrt{\frac{1}{T}\int_0^T |a_i(t)|^2 \mathrm{d}t}$. We define $f(t)$ and $F(\theta)$ to be a rectangle and sinc function respectively:

$$f(t) = \begin{cases} 1 & \text{if } 0 \le t \le T \\ 0 & \text{otherwise} \end{cases}$$

$$F(\theta) = \frac{\sin(2\pi\theta T)}{2\pi\theta}$$

where $F = \widehat{f}$.

$F(\theta)$ has the derivative,

$$F'(\theta) = \frac{2\pi\theta T\cos(2\pi\theta T) - \sin(2\pi\theta T)}{2\pi\theta^2},$$

which means that

$$|F'(\theta)| \lesssim \begin{cases} T^2 & \text{if } \theta \le 1/T \\ T/|\theta| & \text{otherwise} \end{cases}$$

Let $y_i(t) = a_i(t) \cdot f(t)$, then

$$\begin{aligned} \widehat{y}_i(\theta) &= \widehat{a}_i(\theta) * \widehat{f}(\theta) \\ &= \widehat{a}_i(\theta) * F(\theta) \text{ by } F = \widehat{f} \\ &= v_i F(f_i - \theta) - v_i' F(f_i' - \theta) \\ &= (v_i - v_i')F(f_i - \theta) + v_i'(f_i - f_i') \cdot F'(x - \theta) \text{ some } x \in [f_i, f_i']. \end{aligned}$$

We split into two cases. First, if $\nu_i \le \frac{1}{T}$, then

$$\begin{aligned} |\widehat{y}_i(\theta)| &\lesssim (|v_i - v_i'| + \nu_i T|v_i'|) \cdot \min(T, \frac{1}{|f_i - \theta|}) \\ &\lesssim \|a_i\| \cdot \min(T, \frac{1}{|f_i - \theta|}) \quad \text{by Lemma III.1,} \end{aligned} \tag{10}$$

where the first line holds for both $f_i > f_i'$ and $f_i \le f_i'$ since the triangle inequality. Therefore

$$\begin{aligned} & \frac{1}{T}\int_0^T a_i(t)\overline{a_j(t)}\mathrm{d}t \\ = \; & \frac{1}{T}\int_{-\infty}^{\infty} y_i(t)\overline{y_j(t)}\mathrm{d}t \text{ by } y_i(t) = a_i(t) \cdot f(t) \text{ and } f(t) = 1 \text{ iif } t \in [0, T] \\ = \; & \frac{1}{T}\int_{-\infty}^{\infty} \widehat{y}_i(\theta)\overline{\widehat{y}_j(\theta)}\mathrm{d}\theta \text{ by the property of Fourier Transform} \\ \lesssim \; & \frac{1}{T}\|a_i\|\|a_j\| \underbrace{\int_{-\infty}^{+\infty} \min(T, \frac{1}{|f_i - \theta|}) \cdot \min(T, \frac{1}{|f_j - \theta|})\mathrm{d}\theta}_{C} \text{ by Equation (10).} \end{aligned} \tag{11}$$

Using Lemma IV.3, we have following bound for term C,

$$\int_{-\infty}^{+\infty} \min(T, \frac{1}{|f_i - \theta|}) \cdot \min(T, \frac{1}{|f_j - \theta|})\mathrm{d}\theta \lesssim \frac{\log T|f_j - f_i|}{|f_j - f_i|}.$$

This gives the result for $\nu_i \le \frac{1}{T}$. In the alternate case, we have $\nu_i > \frac{1}{T}$, then

$$\begin{aligned} |\widehat{y}_i(\theta)| &\lesssim v_i \cdot \min(T, \frac{1}{|f_i - \theta|}) + v_i' \cdot \min(T, \frac{1}{|f_i' - \theta|}) \\ &\lesssim \|a_i\| \cdot \left(\min(T, \frac{1}{|f_i - \theta|}) + \min(T, \frac{1}{|f_i' - \theta|})\right) \quad \text{by Lemma III.1.} \end{aligned}$$

By similar reason for Equation (11), we have

$$
\begin{aligned}
& \frac{1}{T}\int_0^T a_i(t)\overline{a_j(t)}\mathrm{d}t \\
\lesssim\ & \frac{1}{T}\|a_i\|\|a_j\| \underbrace{\int_{-\infty}^{+\infty} \min(T, \frac{1}{|f_i-\theta|}) \cdot \min(T, \frac{1}{|f_j-\theta|})\mathrm{d}\theta}_{C_1} \\
+\ & \frac{1}{T}\|a_i\|\|a_j\| \underbrace{\int_{-\infty}^{+\infty} \min(T, \frac{1}{|f_i-\theta|}) \cdot \min(T, \frac{1}{|f'_j-\theta|})\mathrm{d}\theta}_{C_2} \\
+\ & \frac{1}{T}\|a_i\|\|a_j\| \underbrace{\int_{-\infty}^{+\infty} \min(T, \frac{1}{|f'_i-\theta|}) \cdot \min(T, \frac{1}{|f_j-\theta|})\mathrm{d}\theta}_{C_3} \\
+\ & \frac{1}{T}\|a_i\|\|a_j\| \underbrace{\int_{-\infty}^{+\infty} \min(T, \frac{1}{|f'_i-\theta|}) \cdot \min(T, \frac{1}{|f'_j-\theta|})\mathrm{d}\theta}_{C_4}.
\end{aligned}
$$

Applying Lemma IV.3 on the term C_1, C_2, C_3 and C_4 respectively,

$$
\frac{1}{T}\int_0^T a_i(t)\overline{a_j(t)} \lesssim \|a_i\|\|a_j\|\frac{\log(T\Delta f_{i,j})}{\Delta f_{i,j}},
$$

where $\Delta f_{i,j} = \min(|f_i - f_j|, |f_i - f'_j|, |f'_i - f_j|, |f'_i - f'_j|)$.

■

Lemma IV.5. *Let $\{(v_i, f_i)\}$ and $\{(v'_i, f'_i)\}$ be two sets of k tones for which $\min_{i\neq j}|f_i - f_j| \geq \eta$ and $\min_{i\neq j}|f'_i - f'_j| \geq \eta$ for some $\eta > 0$. Suppose that $T > C/\eta$ for a sufficiently large constant C. Then these sets can be indexed such that*

$$
\frac{1}{T}\int_0^T |\sum_{i=1}^k (v'_i e^{2\pi \mathbf{i} f'_i t} - v_i e^{2\pi \mathbf{i} f_i t})|^2 \mathrm{d}t \leq (1 + O(\frac{\log(k\eta T)\log(k)}{\eta T}))\sum_{i=1}^k \frac{1}{T}\int_0^T |v'_i e^{2\pi \mathbf{i} f'_i t} - v_i e^{2\pi \mathbf{i} f_i t}|^2 \mathrm{d}t. \tag{12}
$$

Proof: For simplicity, let $a_i(t) = v_i e^{2\pi f_i t \mathbf{i}} - v'_i e^{2\pi f'_i t \mathbf{i}}$. Let's express the square of summations by diagonal term and off-diagonal term, and then bound them separately.

$$
\begin{aligned}
& \int_0^T \left|\sum_{i=1}^k a_i(t)\right|^2 \mathrm{d}t \\
=\ & \int_0^T \left(\sum_{i=1}^k a_i(t)\right)\left(\sum_{i=1}^k \overline{a_i(t)}\right)\mathrm{d}t \\
=\ & \int_0^T \sum_{i=1}^k \underbrace{a_i(t)\overline{a_i(t)}}_{\text{diagonal}} + \sum_{i\neq j}\underbrace{a_i(t)\overline{a_j(t)}}_{\text{off-diagonal}}\mathrm{d}t.
\end{aligned} \tag{13}
$$

Using the result of Lemma IV.4 and $a^2 + b^2 \geq 2ab$, we can upper bound the off-diagonal term,

$$
\begin{aligned}
& \int_0^T a_i(t)\overline{a_j(t)}\mathrm{d}t \\
\lesssim\ & \frac{\log(\Delta f_{i,j}T)}{\Delta f_{i,j}T}\cdot\sqrt{\int_0^T |a_i(t)|^2\mathrm{d}t\int_0^T |a_j(t)|^2\mathrm{d}t} \quad \text{by Lemma IV.4} \\
\lesssim\ & \frac{\log(\Delta f_{i,j}T)}{\Delta f_{i,j}T}\cdot\left(\int_0^T |a_i(t)|^2\mathrm{d}t + \int_0^T |a_j(t)|^2\mathrm{d}t\right) \quad \text{by } 2ab \leq a^2 + b^2.
\end{aligned}
$$

where $\Delta f_{i,j} = \min(|f_i - f_j|, |f_i - f'_j|, |f'_i - f_j|, |f'_i - f'_j|)$. If we index such that any f_i is matched with any f'_j with $|f_i - f'_j| < \eta/3$ – which is possible, since at most one such f'_j will exist by the separation among the f'_j, and that f'_j

will be within $\eta/3$ of at most on f_i – then we have $\Delta f_{i,j} \gtrsim |f_i - f_j|$. If we order the f_i in increasing order, then in fact $\Delta f_{i,j} \gtrsim \eta|i-j|$.

If $T > C/\eta$ for a sufficiently large constant C, this means that $\Delta f_{i,j} T \gtrsim |i-j|\eta T \geq e$. Since $\frac{\log x}{x}$ is decreasing on thie region, this implies

$$\frac{\log(\Delta f_{i,j} T)}{\Delta f_{i,j} T} \lesssim \frac{\log(|i-j|\eta T)}{|i-j|\eta T}.$$

Thus, we have

$$\int_0^T a_i(t)\overline{a_j(t)}\mathrm{d}t \lesssim \frac{\log(|i-j|\eta T)}{|i-j|\eta T} \cdot \left(\int_0^T |a_i(t)|^2 \mathrm{d}t + \int_0^T |a_j(t)|^2 \mathrm{d}t \right) \quad . \tag{14}$$

Finally, we have

$$\begin{aligned}
& \int_0^T \left| \sum_{i=1}^k a_i(t) \right|^2 \mathrm{d}t - \sum_{i=1}^k \int_0^T |a_i(t)|^2 \mathrm{d}t \\
= & \sum_{i \neq j}^k \int_0^T a_i(t)\overline{a_j(t)}\mathrm{d}t \quad \text{by Equation (13)} \\
\lesssim & \sum_{i \neq j}^k \frac{\log(|i-j|\eta T)}{|i-j|\eta T} \cdot \left(\int_0^T |a_i(t)|^2 \mathrm{d}t + \int_0^T |a_j(t)|^2 \mathrm{d}t \right) \quad \text{by Equation (14)} \\
\leq & \frac{\log(k\eta T)}{\eta T} \sum_{i=1}^k \sum_{j \neq i}^k \frac{1}{|i-j|} \cdot \left(\int_0^T |a_i(t)|^2 \mathrm{d}t + \int_0^T |a_j(t)|^2 \mathrm{d}t \right) \\
= & 2\frac{\log(k\eta T)}{\eta T} \sum_{i=1}^k \sum_{j \neq i}^k \frac{1}{|i-j|} \int_0^T |a_i(t)|^2 \mathrm{d}t \quad \text{by symmetry} \\
\lesssim & \frac{\log(k\eta T)}{\eta T} \sum_{i=1}^k \int_0^T |a_i(t)|^2 \mathrm{d}t \sum_{j \neq i}^k \frac{1}{|i-j|} \\
\lesssim & \frac{\log(k\eta T)\log(k)}{\eta T} \sum_{i=1}^k \int_0^T |a_i(t)|^2 \mathrm{d}t \quad \text{by } \sum_{i=1}^k \frac{1}{i} \approx \log(k) \ .
\end{aligned}$$

Thus, we complete the proof. ∎

Lemma III.13. *Let $\{(v_i, f_i)\}$ and $\{(v'_i, f'_i)\}$ be two sets of k tones for which $\min_{i \neq j} |f_i - f_j| \geq \eta$ and $\min_{i \neq j} |f'_i - f'_j| \gtrsim \eta$ for some $\eta > 0$. Suppose that $T > O(\frac{\log^2 k}{\eta})$. Then these sets can be indexed such that*

$$\frac{1}{T} \int_0^T |\sum_{i=1}^k (v'_i e^{2\pi \mathbf{i} f'_i t} - v_i e^{2\pi \mathbf{i} f_i t})|^2 \mathrm{d}t \lesssim \sum_{i=1}^k \frac{1}{T} \int_0^T |v'_i e^{2\pi \mathbf{i} f'_i t} - v_i e^{2\pi \mathbf{i} f_i t}|^2 \mathrm{d}t. \tag{15}$$

Proof: Directly follows by Lemma IV.5. ∎

V. Lower Bound

Lemma III.15. *There exists a constant $c > 0$ such that, for a given sample duration T, one cannot recover the frequency f to within*

$$c\frac{\mathcal{N}}{T|\widehat{x}^*(f)|}$$

with $3/4$ probability, for all $\delta > 0$, even if $k = 1$.

Proof:

Suppose this were possible, and consider two one-sparse signals y and y' containing tones (v, f) and (v, f'), respectively. By Lemma III.1,

$$\int_0^T |y(t) - y(t)|^2 \mathrm{d}t \lesssim |v|^2 T^2 |f - f'|^2.$$

Consider recovery of the signal $x(t) = y(t)$, and suppose it outputs some frequency f^*. This must simultaneously be a good recovery for the decomposition $(x^*, g) = (y, 0)$ and $(x^*, g) = (y', y - y')$. These have noise levels $\mathcal{N}^2$ bounded by $\delta|v|^2$ and $\delta|v|^2 + O(|v|^2T^2|f - f'|^2)$, respectively. By the assumption of good recovery, and the triangle inequality, we require

$$c\frac{2\sqrt{\delta|v|^2} + \sqrt{O(|v|^2T^2|f - f'|^2)}}{Tv} \gtrsim |f - f'|$$

or

$$c \cdot O(\sqrt{\frac{\delta}{T|f - f'|}} + 1) \geq 1.$$

Because δ may be chosen arbitrarily small, we can choose a small constant c such that this is a contradiction. ■

Algorithm 1 Continuous Fourier Sparse Recovery

procedure ContinuousFourierSparseRecovery($x, k, \delta, \alpha, C, F, T, \eta$) —— The Main Algorithm
  F is the upper bound of frequency, T is the sample duration, C is the approximation factor.
  δ, α are the parameters associated with Hash function.
  $c = 1/10$ and $b = 8/10$
  $R_1 \leftarrow$ NoisyKSparseCFFT$(x, k, \delta, \alpha, C, F)$
  $S_1 \leftarrow$ MergedStages$(R_1, O(k \log k), \eta, c, b)$
  $R_2 \leftarrow$ NoisyKSparseCFFT$(x, O(k), \delta, \alpha, C, F, T)$
  $S_2 \leftarrow$ MergedStages$(R_2, O(k \log k), \Omega(\eta), c, b)$
  $S \leftarrow S_1 \cap S_2$, which means only keeping the tones that S_1 agrees with S_2 by Lemma III.10.
  $S^* \leftarrow$ Prune(S, k), which means only keeping the top-k largest magnitude tones.
  return S^*
end procedure
procedure NoisyKSparseCFFT$(x, k, \delta, \alpha, C, F, T)$
  Let $B = k/\epsilon$.
  for $c = 1 \to \log(k)$ **do**
    Choose σ uniformly at random from $[\frac{1}{B\eta}, \frac{2}{B\eta}]$.
    Choose b uniformly at random from $[0, \frac{2\pi\lceil F/\eta \rceil}{\sigma B}]$.
    $R_c \leftarrow$ OneStage$(x, B, \delta, \alpha, \sigma, b, C, F, T)$
  end for
  return $(R_1, R_2, \cdots, R_{\log(k)})$.
end procedure
procedure MergedStages(R, m, η, c, b)
  R is a list of m tones (v_i', f_i')
  c is some constant < 1.
  b is some constant < 1.
  Sort list R based on f_i'.
  Building the 1D range search Tree based on m points by regarding each frequency f_i' as a 1D point on a line where $x_i = f_i'$.
  $S \leftarrow \emptyset$, $i \leftarrow 0$
  while $i < m$ **do**
    if Tree.Count$(f_i', f_i' + c\eta) \geq b \log k$ **then**
      $f \leftarrow$ median $\{ f_j' \mid f_j' \in [f_i' - c\eta, f_i' + 2c\eta]\}$
      $v \leftarrow$ median $\{ v_j' \mid f_j' \in [f_i' - c\eta, f_i' + 2c\eta]\}$
      $S \leftarrow S \cup (f, v)$
      $i \leftarrow$ Tree.Search$(f_i' + 2c\eta + \eta/2)$, which means walk to the first point that is on the right of $f_i' + 2c\eta + \eta/2$
    else
      $i \leftarrow i + 1$
    end if
  end while
  return S
end procedure

Algorithm 2 Continuous Fourier Sparse Recovery

procedure HashToBins($x, P_{\sigma,a,b}, B, \delta, \alpha$)
  Compute $\widehat{y}_{jF/B}$ for $j \in [B]$, where $y = G_{B,\alpha,\delta} \cdot (P_{\sigma,a,b}x)$
  return $\widehat{u}$ given by $\widehat{u}_j = \widehat{y}_{jF/B}$
end procedure
procedure OneStage($x, B, \delta, \alpha, \sigma, b, C, F, T$)
  $L \leftarrow$ LocateKSignal($x, B, \delta, \alpha, \sigma, b, C, F, T$)
  Choose $a \in [0, 1]$ uniformly at random.
  $\widehat{u} \leftarrow$ HashToBins($x, P_{\sigma,a,b}, B, \delta, \alpha$)
  return $\{(\widehat{u}_{h_{\sigma,b}(f')} e^{-2\pi\sigma a f' \mathbf{i}}, f')$ for $f' \in L$ if not $E_{off}(f')\}$.
end procedure
procedure LocateKSignal($x, B, \delta, \alpha, \sigma, b, C, F, T$)
  Set $t \approx \log(FT)$, $D \approx \log_t(FT)$, $R_{loc} \approx \log_C(tC)$, $l^{(1)} = F/2$.
  for $i \in [D-1]$ **do**
    $\Delta l = F/(t)^{i-1}$, $s = \frac{1}{\sqrt{C}}$, $\widehat{\beta} = \frac{ts}{2\sigma\Delta l}$
    $l^{(i+1)} \leftarrow$ LocateInner($x, B, \delta, \alpha, \sigma, b, \widehat{\beta}, l^{(i)}, \Delta l, t, R_{loc}$, **false**).
  end for
  Set $s = 1/C$, $t \approx \log(FT)/s$, $\Delta l \approx st/T$, $\widehat{\beta} = \frac{ts}{2\sigma\Delta l}$, $R_{loc} \approx \log_C(tC)$
  $l^{(*)} \leftarrow$ LocateInner($x, B, \delta, \alpha, \sigma, b, \widehat{\beta}, l^{(D)}, \Delta l, t, R_{loc}$, **true**).
  return $l^{(*)}$.
end procedure
procedure LocateInner($x, B, \delta, \sigma, b, \widehat{\beta}, l, \Delta l, t, R_{loc}, last$)
  Let $v_{j,q} = 0$ for $(j, q) \in [B] \times [t]$.
  for $r \in [R_{loc}]$ **do**
    Choose $\gamma \in [\frac{1}{2}, 1]$ uniformly at random.
    Choose $\beta \in [\frac{1}{2}\widehat{\beta}, 1\widehat{\beta}]$ uniformly at random.
    $\widehat{u} \leftarrow$ HashToBins($x, P_{\sigma,\gamma,b}, B, \delta, \alpha$).
    $\widehat{u}' \leftarrow$ HashToBins($x, P_{\sigma,\gamma+\beta,b}, B, \delta, \alpha$).
    for $j \in [B]$ **do**
      for $i \in [m]$ **do**
        $\theta^r_{j,i} = \frac{1}{2\pi\sigma\beta}(\phi(\widehat{u}_j/\widehat{u'}_j) + 2\pi s_i), s_i \in [\sigma\beta(l_j - \Delta l/2), \sigma\beta(l_j + \Delta l/2)] \cap \mathbb{Z}_+$
        $f^r_{j,i} = \theta^r_{j,i} + b \pmod F$
        suppose $f^r_{j,i}$ belongs to region(j, q),
        add a vote to both region(j, q) and two neighbors nearby that region, e.g. region$(j, q-1)$ and region$(j, q+1)$
      end for
    end for
  end for
  for $j \in [B]$ **do**
    $q^*_j \leftarrow \{q | v_{j,q} > \frac{R_{loc}}{2}\}$
    if $last$ = **true then**
      $l^*_j \leftarrow$ median$\{f^r_{j,i} | f^r_{j,i} \in$ region$(j, q^*_j), i \in [f], r \in [R_{loc}]\}$
    else
      $l^*_j \leftarrow$ center of region(j, q^*_j)
    end if
  end for
  return l^*
end procedure