MIT 6.893: Algorithms and Signal Processing	Fall 2014

Lecture 2 — September 10, 2014

Prof. Piotr Indyk — *Scribe: Christopher Musco*

1 Sparse Fourier Tranform, $k > 1$

Last lecture, we discussed algorithms for computing a good 1-sparse approximation to the Fourier transform of a signal. In this lecture, we generalize to the k-sparse case so that we can handle signals with multiple large Fourier coefficients. Our approach will reduce the problem to the 1-sparse case and then use our previous 2-point sampling algorithm black-box (we will not apply the noise robust 1-sparse algorithm yet).

This is essentially accomplished by isolating large coefficients. Today, we will isolate coefficients via aliasing, but we will see alternative approaches in later lectures. In general, an aliasing approach requires strong assumptions on our signal – we need the locations of Fourier peaks in $\hat{\mathbf{a}}$ to be randomly generated. Nevertheless, the algorithm presented is easily implemented and quite useful in practice.

2 Aliasing

The essential idea behind aliasing is that *down sampling* in the time domain leads to an aliased Fourier spectrum. This lets us deal with fewer time samples. However, even if our signal is sparse, there is a chance that aliasing will cause some Fourier components to collapse together, thus making them impossible to recover. Nevertheless, we claim this does not happen too often if our signal is randomly generated.

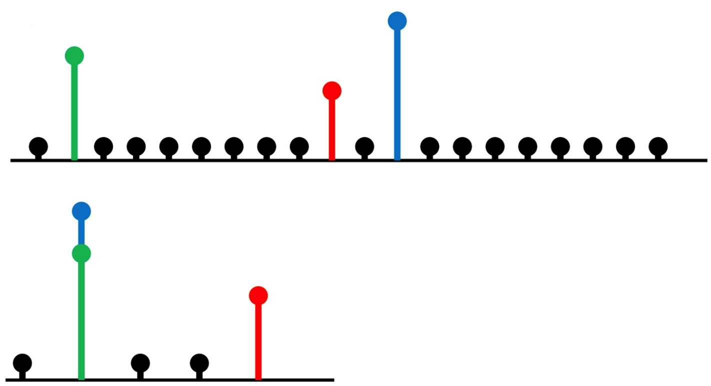

Figure 1: Aliasing of a sparse signal that leads to overlapping components

Theorem 1 (Aliasing Theorem). *Consider a signal* $\mathbf{a} = \{a_0, \ldots, a_{n-1}\}$ *of length* n *and a down-sampling parameter* L *that evenly divides* n*. Let* $\mathbf{a}' = \{a_0, a_L, a_{2L}, \ldots\}$*. Then*

$$\mathbf{F}\mathbf{a}'(u) = \hat{a}'_u = \sum_{l=0}^{L-1} \hat{a}_{u+l\frac{n}{L}} \tag{1}$$

Proof. Using our chosen normalization of the Fourier transform from last lecture, for any k,

$$\hat{a}_k = \frac{1}{n}\sum_{j=0}^{n-1} a_j e^{-\frac{2\pi i}{n}kj}.$$

Plugging into the right hand side of our theorem gives:

$$\begin{aligned}
\sum_{l=0}^{L-1} \hat{a}_{u+l\frac{n}{L}} &= \sum_{l=0}^{L-1}\frac{1}{n}\sum_{j=0}^{n-1} a_j e^{-\frac{2\pi i}{n}(u+l\frac{n}{L})j} \\
&= \frac{1}{n}\sum_{j=0}^{n-1}\sum_{l=0}^{L-1} a_j e^{-\frac{2\pi i}{n}(u+l\frac{n}{L})j} \\
&= \frac{1}{n}\sum_{j=0}^{n-1} a_j e^{-\frac{2\pi i}{n}(uj)} \sum_{l=0}^{L-1} e^{-2\pi i(\frac{lj}{L})}. \qquad (2)
\end{aligned}$$

Now just consider the interior sum:

$$\sum_{l=0}^{L-1}\left(e^{-2\pi i(\frac{j}{L})}\right)^l.$$

When $(j \bmod L) = 0$, j/L is a whole number so $e^{-2\pi i(\frac{j}{L})}$ evaluates to 1. Thus, this sum is simply equal to L. When $(j \bmod L) \neq 0$, the sum is actually equal to 0. One way to see this is through the closed form solution of a complex geometric series:

$$\frac{1-\left(e^{-2\pi i(\frac{j}{L})}\right)^L}{1-e^{-2\pi i(\frac{j}{L})}} = \frac{1-e^{-2\pi ij}}{1-e^{-2\pi i(\frac{j}{L})}}$$

The top of the fraction equals 0 and when j/L is not an integer, the bottom is some non-zero complex number, so the whole expression evaluates to zero.

So, the interior sum effectively samples from the exterior sum. Plugging into (2), we see that

$$\begin{aligned}
\sum_{l=0}^{L-1} \hat{a}_{u+l\frac{n}{L}} &= \frac{1}{n}\sum_{j=0,(j \bmod L)=0}^{n-1} a_j e^{-\frac{2\pi i}{n}(uj)} \times L \\
&= \frac{1}{n/L}\sum_{t=0}^{n/L-1} a_{Lt} e^{-\frac{2\pi i}{n/L}(ut)}
\end{aligned}$$

This expression is simply equivalent to $\mathbf{Fa}'(u)$. □

3 Isolating Components

As mentioned, aliasing allows us to compute a Fourier transform on a significantly subsampled version of our signal. If our original spectrum is not adversarial (i.e. no frequencies spaced apart by an exact multiple of n/L) the spectrum of the sampled signal should contain all of the information from the original. This should hold with high probability for a randomly generated spectrum. The theorem corresponding our first (noiseless) k-sparse algorithm is:

Theorem 2 (k-Sparse Algorithm)**.** *Suppose that* $\hat{\mathbf{a}}$ *is generated by summing up* k *Fourier coefficients in positions selected independently and uniformly at random from* $\{0, \ldots, n-1\}$*. I.e.*

$$\hat{\mathbf{a}} = \sum_{i=1}^{k} b_t \delta_{j_t},$$

where $b_1, \ldots, b_k$ *are fixed magnitudes,* $j_1, \ldots, j_k$ *are i.i.d random variables chosen uniformly from* $\{0, \ldots, n-1\}$*, and* δ *is the Dirac function. Then, given* $\mathbf{a}$*, we can recover* $\hat{\mathbf{a}}$ *in* $O(k^2 \log k)$ *time with probability at least* $1/2$*.*

This is an *average case* result since the algorithm will not work for an adversarially chosen signal. Nevertheless, in future lectures we will see that worst case sFFT algorithms are also possible.

First, note that we assume $k^2 \ll n$ – otherwise we may as well compute a full FFT in $n \log n$ time. Then, the algorithm is as follows:

1. Set $L = n/k^2$
2. Define $\mathbf{a}' = \{a_0, a_L, a_{2L}, \ldots\}$ and $\mathbf{a}'' = \{a_1, a_{L+1}, a_{2L+1}, \ldots\}$. Both sampled signals have length k^2.
3. Compute $\hat{\mathbf{a}}'$ and $\hat{\mathbf{a}}''$ using a standard FFT. Recall from Theorem 1 and the time shift theorem that:

$$\hat{a}'_u = \sum_{l=0}^{L-1} \hat{a}_{u+l\frac{n}{L}}, \tag{3}$$

$$\hat{a}''_u = \sum_{l=0}^{L-1} \hat{a}_{u+l\frac{n}{L}} \omega^{u+l\frac{n}{L}} \tag{4}$$

For each non-zero in $\hat{\mathbf{a}}'$ apply the two-point sampling algorithm from Lecture 1. Recall that we can recover $u + l\frac{n}{L}$ (the index of the non-zero coordinate in $\mathbf{a}$) from $\hat{a}''_u/\hat{a}'_u$.

The analysis of this algorithm is simple. It succeeds as long as each non-zero coordinate in $\hat{\mathbf{a}}'$ only receives a contribution from a single Fourier component in $\hat{\mathbf{a}}$. The probability that this does not occur is bounded by $\frac{k(k-1)}{2} \times \frac{L}{n}$ – i.e. the total number of index pairs times the probability that each pair collides when down sampling by L. Since we chose L such that $\frac{1}{k^2} = \frac{n}{L}$ this failure probability is upper bounded by $1/2$. Thus, we succeed with at least $1/2$ probability.

The algorithm described requires two FFTs of length k^2 and then at most k $O(1)$ operations to recover the indices of our sparse spectrum. In total, its running time is $O(k^2 \log k)$. In the next lecture, we will see how to decrease the k^2 dependence to k.