Notes on Lemma 6, July 26, 2012
Mark Iwen

Lemma 6 on page 19 in Section 6 of “Combinatorial Sublinear-Time Fourier Algorithms” contains a couple typos, one of which necessitates a correction. In particular, the calculation performed on line seven of the proof omits a constant factor of \((2\pi)^{2\kappa}\) which, when corrected, has the unfortunate effect of causing the interpolation error considered therein to grow exponentially, instead of shrinking exponentially as desired. This omission is ultimately a consequence of mistakenly considering \(p^*_{\kappa}\) as a polynomial with domain \([0,1]\) instead of \([0,2\pi]\).

It is worth mentioning that the current proof works as written for signals which are oversampled by a factor of \(\geq 2\). That is, if \(\hat{A}(\omega) = 0\) for all \(|\omega| > N/4\), then Lemma 6 holds essentially as stated. In fact, the neglected constant factor of \((2\pi)^{2\kappa}\) can be entirely erased by considering the signal in question to be oversampled by a factor of \(2\pi\) (i.e., the oversampling rate directly cancels the neglected constant). This oversampling factor can be reduced to 2 if one is willing to accept a slower rate of exponential decay in interpolation error as \(\kappa\) increases. However, fixing Lemma 6 so that it works for general trigonometric polynomials of degree \(N/2\) requires more work.

1 Correction in the General Case

In order to correct Lemma 6 in general we will consider a modified version of the function, \(f : [0,2\pi] \rightarrow \mathbb{C}\), defined at the bottom of page 18. Instead, we replace \(f\) with the closely related function \(f_0\) defined by

\[
f_0(x) := \frac{1}{2\pi} \sum_{\omega=1-\left\lfloor \frac{N}{2} \right\rfloor}^{\left\lfloor \frac{N}{2} \right\rfloor} \hat{G}(\omega) \hat{A}(\omega) \cdot e^{i\omega \cdot x}, \quad x \in [0,2\pi].
\]

(1)

Here \(\hat{G}\) is the sequence of Fourier coefficients of a filter function. Thus, \(f_0\) is effectively a filtered version of \(f\). We imagine that we have sampling access to \(f\) as defined on page 18, but do our calculations with the filtered version, \(f_0\), instead. We will discuss the discrete and (effectively) band limited “low-pass” filter, \(G\), in more detail below (see Section 2). For now, we simply assume that there exists a constant \(C \in \mathbb{R}^+\) such that \(|\hat{G}(\omega)| \leq C \cdot e^{-|\omega| \cdot 8\kappa / N}\) for all \(\omega \in \mathbb{Z}\).

Correcting the calculation on line seven of the proof of Lemma 6 by inserting the omitted

\footnote{I would like to thank Jieming Mao for bringing my attention to the error discussed herein.}
constant factor in red, we note that:

\[ |\Re \{ f_0(x) \} - \hat{p}_R^x(x) | \leq \frac{\|f_0^{(2k)}\|_\infty}{(2\kappa)!} \cdot \prod_{m=1}^{\kappa} \left( \frac{m \cdot 2\pi}{N} \right)^2. \]  

(2)

We must now correct for this additional constant factor, which is accomplished in the general case by replacing \( f \) with \( f_0 \) in the calculation. Note that

\[
\left\| f_0^{(2k)} \right\|_\infty \leq \frac{C}{2\pi} \sum_{\omega=1-\left\lfloor \frac{N}{2} \right\rfloor}^{\frac{N}{2}} |\omega|^{2\kappa} \cdot e^{-|\omega| \cdot 8\kappa / N} \cdot \left| \hat{\mathbf{A}}(\omega) \right| \leq \frac{C}{2\pi} \left( \frac{N}{4e} \right)^{2\kappa} \sum_{\omega=1-\left\lfloor \frac{N}{2} \right\rfloor}^{\frac{N}{2}} \left| \hat{\mathbf{A}}(\omega) \right|.
\]

Continuing our calculation from Equation 2 we see that

\[
|\Re \{ f_0(x) \} - \hat{p}_R^x(x) | \leq \frac{1}{(2\kappa)!} \frac{C \cdot \| \hat{\mathbf{A}} \|_1}{2\pi} \left( \frac{N}{4e} \right)^{2\kappa} \prod_{m=1}^{\kappa} \left( \frac{m \cdot 2\pi}{N} \right)^2 \leq \frac{C \cdot \| \hat{\mathbf{A}} \|_1}{2\pi \cdot 2^\kappa} \cdot \prod_{m=1}^{\kappa} m^2 \frac{2^\kappa}{(2\kappa)!}.
\]

Thus, \( |\Re \{ f_0(x) \} - \hat{p}_R^x(x) | \leq \frac{C \cdot \| \hat{\mathbf{A}} \|_1}{2\pi \cdot 2^\kappa} \). The remainder of the argument goes through as before, and we obtain the following modified form of Lemma 6.

**Lemma 6.** Let \( \mathbf{A} \) be an \( N \)-length complex valued array and suppose that \( \hat{\mathbf{A}} \) is \((c,p)\)-compressible. Fix \( \bar{c} \in \mathbb{R}^+ \). Using \( 2\kappa = O(\log(p \cdot \kappa / \bar{c} \cdot \delta)) \) interpolation points from \( \mathbf{A} \) per \( f_0 \)-evaluation will guarantee that every line 8 DFT entry from Algorithm 2 is calculated to within \( \bar{c} \kappa \cdot 2^p \cdot (2\kappa)! \) precision.

We conclude this section by pointing out that this modified version of Lemma 6 can still be used to prove a modified version of Corollary 5 on page 20. This can be done by executing Algorithm 2 \( O(\kappa) \)-times on \( O(\kappa) \) different \( f_0 \) variants, instead of executing it on \( f \) directly. Given that \( \hat{\mathbf{G}} \) is known and relatively large for all \( \omega \) with \( |\omega| = O(N/\kappa) \), we can recover all energetic frequencies of size \( O(N/\kappa) \) from \( \hat{\mathbf{A}} \) by using the results of Algorithm 2 on \( f_0 \) (see Equation 1 for the definition of \( f_0 \)). Thus, we can recover all energetic frequencies from \( \mathbf{A} \) by modulating \( f \) \( O(\kappa) \)-times and then filtering with \( \mathbf{G} \). In particular, we may define

\[
f_{j'}(x) := \left( \mathbf{G} \ast e^{i \cdot 2\pi x [j' N / C'' \kappa]} f \right)(x) \approx \frac{1}{2\pi} \sum_{\omega=1-\left\lfloor \frac{N}{2} \right\rfloor}^{\frac{N}{2}} \hat{\mathbf{G}}(\omega) \hat{\mathbf{A}} \left( \omega \left[ j' N / C'' \kappa \right] \right) \cdot e^{i \omega x}, \quad x \in [0, 2\pi],
\]

for \( j' \in [-C'' \kappa, C'' \kappa] \cap \mathbb{Z} \) and fixed constants \( C', C'' \in \mathbb{N} \). Executing Algorithm 2 on each of these \( f_{j'} \) will allow one to recover all energetic frequencies from \( \mathbf{A} \).

2 The filter \( \mathbf{G} \)

The filter \( \mathbf{G} \) must have several properties in order to allow the production of a sublinear-time Fourier algorithm (i.e., in order for a modified version of Corollary 5 on page 20 to
hold as discussed above). Most important among these properties are the following: First, the filter array $G : [1, N] \cap \mathbb{Z} \rightarrow \mathbb{C}$ should be zero almost everywhere. This allows $f_j' = G \ast \left( e^{i x \cdot \frac{N}{C' \kappa}} f \right)$ to be sampled quickly using only a few samples from $f$ in the process. Of course, it is much more likely that $G$ will be “almost zero” everywhere, in which case the convolution involved in the definition of $f_j'$ can still be (approximately) computed both quickly and accurately using only a few samples from $f$.

Second, the Fourier transform of the filter, $\hat{G}$, should have both the properties alluded to in Section 1 above. Mainly, $\hat{G}$ should exhibit exponential decay for larger frequencies, i.e. $|\hat{G}(\omega)|$ should be $O(e^{-|\omega| \cdot 8\kappa/N})$. However, $|\hat{G}(\omega)|$ should not decay too quickly. That is, $G$ should serve as a decent low-pass filter. In particular, we require that $|\hat{G}(\omega)|$ be relatively large (e.g., larger than $1/10$) for all $\omega$ with $|\omega| \leq N/2\kappa$.

Gaussian filters generally fulfill the required properties listed above. For example, one can take

$$\hat{G}(\omega) = \begin{cases} e^{-\frac{2}{N^2} \omega^2} & \text{if } \omega \in \left(\left\lceil \frac{N}{2} \right\rceil, \left\lfloor \frac{N}{2} \right\rfloor \right) \cap \mathbb{Z} \\ 0 & \text{otherwise} \end{cases}.$$  

The filter $G$ can then be taken as the inverse discrete Fourier transform of $\hat{G}$.$^2$ In this case, $G$ will also “look Gaussian”, and therefore be “almost zero everywhere” as desired. See Figure 1 for graphs of this Gaussian filter when $N = 200,001$ and $\kappa = 7$. Note that the desired properties we have discussed in this section are indeed achieved in this example.

$^2$Creating $G$ in this fashion will result in a one-time computational cost of $O(N \log N)$. This one-time cost can be avoided however – see, e.g., section 7 of “Nearly Optimal Sparse Fourier Transform” by Hassanieh, Indyk, Katabi, and Price.
Figure 1: The example filter, \( \mathbf{G} \), in Equation 3 with length \( N = 200,001 \) and \( \kappa = 7 \). The top graph demonstrates that \( |F[\mathbf{G}]| = |\hat{\mathbf{G}}| \) decays exponentially in accordance with the assumption made in Section 1 above. Furthermore, \( F[\mathbf{G}] = \hat{\mathbf{G}} \) is shown to be relatively large in magnitude (e.g., larger than 0.1) for all frequencies \( \omega \) with \( |\omega| \leq 20,000 \). Hence, the filter effectively passes one fifth of the lowest magnitude frequencies. The bottom graph demonstrates the exponential decay of the entries of \( \mathbf{G} \) in magnitude. Hence, convolutions with \( \mathbf{G} \) can be approximatly sampled both quickly and accurately.