PAMM · Proc. Appl. Math. Mech. **15**, 667 – 668 (2015) / **DOI** 10.1002/pamm.201510323

Deterministic sparse FFT algorithms

Gerlind Plonka[1,*] and **Katrin Wannenwetsch**[1]

[1] Institut für Numerische und Angewandte Mathematik, Georg-August-Universität Göttingen, D-37083 Göttingen, Germany

In this paper we consider sparse signals $\mathbf{x} \in \mathbf{C}^N$ which are known to vanish outside a support interval of length bounded by $m < N$. For the case that m is known, we propose a deterministic algorithm of complexity $\mathcal{O}(m \log m)$ for reconstruction of $\mathbf{x}$ from its discrete Fourier transform $\widehat{\mathbf{x}} \in \mathbf{C}^N$.

1 Introduction

Fast algorithms for the computation of the discrete Fourier transform of a vector of length N have been known for many years. These FFT algorithms have an algorithmic complexity of $\mathcal{O}(N \log N)$. Recently, there has been a stronger interest in Fourier algorithms for sparse vectors which can even achieve a sublinear complexity. Randomized sparse Fourier algorithms achieving a complexity of $\mathcal{O}(m \log N)$ resp. $\mathcal{O}(m \log m)$ for m-sparse vectors can e.g. be found in [2] resp. [3], [4]. An overview of the methods of randomized sparse Fourier transforms is given in [1].

In this paper, we present a deterministic FFT algorithm and restrict ourselves to vectors with a short support interval. Such vectors occur in different applications, such as in X-ray microscopy, where compact support is a frequently used a-priori condition in phase retrieval, as well as in computer tomography reconstructions.

Let $\mathbf{x} \in \mathbf{C}^N$. We define the *support length* $m = |\mathrm{supp}\, \mathbf{x}|$ of $\mathbf{x}$ as the minimal integer m for which there exists a $\mu \in \{0, \ldots, N-1\}$ such that the components x_k of $\mathbf{x}$ vanish for all $k \notin I := \{(\mu + r) \mod N, \quad r = 0, \ldots, m-1\}$. The index set I is called *support interval* of $\mathbf{x}$. We always have $x_\mu \neq 0$ and $x_{\mu+m-1} \neq 0$, but there may be zero components of $\mathbf{x}$ within the support interval. Observe that if $m \leq \frac{N}{2}$, the support interval and hence the first support index μ of $\mathbf{x}$ is uniquely determined.

We define the discrete Fourier transform of a vector $\mathbf{x} \in \mathbf{C}^N$ by $\widehat{\mathbf{x}} = \mathbf{F}_N \mathbf{x}$, where the Fourier matrix $\mathbf{F}_N$ is given by $\mathbf{F}_N := (\omega_N^{jk})_{j,k=0}^{N-1}$, $\omega_N := \mathrm{e}^{-\frac{2\pi \mathrm{i}}{N}}$. In the following, we describe a deterministic algorithm for the reconstruction of $\mathbf{x}$ of length $N = 2^J$ from Fourier data $\widehat{\mathbf{x}} \in \mathbf{C}^N$. The algorithm is based on the idea that the (at most) m nonzero components of $\mathbf{x}$ can already be identified from a periodization of $\mathbf{x}$ of length $2^L \geq m$. Hence for the complete reconstruction it remains to determine the support interval (i.e., the first support index) of $\mathbf{x}$.

2 Reconstruction of $\mathbf{x}$ with short support interval

Let $N := 2^J$ for some $J > 0$. We define the periodizations $\mathbf{x}^{(j)} \in \mathbf{C}^{2^j}$ of $\mathbf{x}$ by

$$\mathbf{x}^{(j)} = (x_k^{(j)})_{k=0}^{2^j-1} = \left(\sum_{\ell=0}^{2^{J-j}-1} x_{k+2^j\ell} \right)_{k=0}^{2^j-1} \tag{1}$$

for $j = 0, \ldots, J$. Obviously, $\mathbf{x}^{(0)} = \sum_{k=0}^{N-1} x_k$ is the sum of all components of $\mathbf{x}$, $\mathbf{x}^{(1)} = (\sum_{k=0}^{N/2-1} x_{2k}, \sum_{k=0}^{N/2-1} x_{2k+1})^T$ and $\mathbf{x}^{(J)} = \mathbf{x}$. The discrete Fourier transform of the vectors $\mathbf{x}^{(j)}$, $j = 0, \ldots, J$, can be described in terms of $\widehat{\mathbf{x}}$. According to the following lemma, it can be obtained by just picking suitable components of $\widehat{\mathbf{x}}$.

Lemma 2.1 *For the vectors* $\mathbf{x}^{(j)} \in \mathbf{C}^{2^j}$, $j = 0, \ldots, J$, *in* (1)*, we have the discrete Fourier transform*

$$\widehat{\mathbf{x}}^{(j)} := \mathbf{F}_{2^j} \mathbf{x}^{(j)} = (\widehat{x}_{2^{J-j}k})_{k=0}^{2^j-1},$$

where $\widehat{\mathbf{x}} = (\widehat{x}_k)_{k=0}^{N-1} = \mathbf{F}_N \mathbf{x}$ *is the Fourier transform of* $\mathbf{x} \in \mathbf{C}^N$.

Assume that the Fourier data $\widehat{\mathbf{x}} = \mathbf{F}_N \mathbf{x} \in \mathbf{C}^N$ and $|\mathrm{supp}\, \mathbf{x}| \leq m$ for some given m. Choose L such that $2^{L-1} < m \leq 2^L$. By Lemma 2.1 we have $\widehat{\mathbf{x}}^{(L+1)} = (\widehat{x}_{2^{J-(L+1)}k})_{k=0}^{2^{L+1}-1}$. Thus, we can compute $\mathbf{x}^{(L+1)}$ using inverse FFT of length 2^{L+1}.

The resulting vector $\mathbf{x}^{(L+1)}$ has already the same support length as $\mathbf{x}$, since $|\mathrm{supp}\, \mathbf{x}| \leq m \leq 2^L$, and for each $k \in \{0, \ldots, 2^{L+1}-1\}$ the sum in

$$x_k^{(L+1)} = \sum_{\ell=0}^{2^{J-L-1}-1} x_{k+2^{L+1}\ell} \tag{2}$$

* Corresponding author: e-mail plonka@math.uni-goettingen.de

contains at most one nonvanishing term. Therefore, the support of $\mathbf{x}^{(L+1)}$ and its first index $\mu^{(L+1)}$ are uniquely determined. For reconstruction of the complete vector $\mathbf{x}$ it is now sufficient to determine the first support index $\mu^{(J)} = \mu$ of the support interval of $\mathbf{x}$. Then the components of $\mathbf{x}$ are given by

$$x_{(\mu^{(J)}+k) \bmod N} = \begin{cases} x^{(L+1)}_{(\mu^{(L+1)}+k) \bmod 2^{L+1}} & k = 0, \ldots, m-1, \\ 0 & k = m, \ldots, N-1. \end{cases} \tag{3}$$

By the following theorem (cf. Theorem 3.1 in [5]), it is possible to obtain $\mu^{(J)}$ and hence to recover $\mathbf{x}$ from the vector $\mathbf{x}^{(L+1)}$ and one additional Fourier component.

Theorem 2.2 *Let $\mathbf{x} \in \mathbb{C}^N$, $N = 2^J$, have support length m (or a support length bounded by m) with $2^{L-1} < m \leq 2^L$. For $L < J-1$, let $\mathbf{x}^{(L+1)}$ be the 2^{L+1}-periodization of $\mathbf{x}$. Then $\mathbf{x}$ can be uniquely recovered from $\mathbf{x}^{(L+1)}$ and one nonzero component of the vector $(\widehat{x}_{2k+1})_{k=0}^{N/2-1}$.*

3 Sparse FFT Algorithm

We summarize the reconstruction of $\mathbf{x}$ from Fourier data $\widehat{\mathbf{x}}$ in the following algorithm.

Algorithm 3.1 *(Sparse FFT for vectors with short support)*

Input: $\widehat{\mathbf{x}} \in \mathbb{C}^N$, $N = 2^J$, $|\mathrm{supp}\,\mathbf{x}| \leq m < N$.

- *Compute L such that $2^{L-1} < m \leq 2^L$, i.e., $L := \lceil \log_2 m \rceil$.*
- *If $L = J$ or $L = J-1$, compute $\mathbf{x} = \mathbf{F}_N^{-1}\widehat{\mathbf{x}}$ using an FFT of length N.*
- *If $L < J-1$:*

1. *Choose $\widehat{\mathbf{x}}^{(L+1)} := (\widehat{x}_{2^{J-(L+1)}k})_{k=0}^{2^{L+1}-1}$ and compute $\mathbf{x}^{(L+1)} := \mathbf{F}_{2^{L+1}}^{-1}\widehat{\mathbf{x}}^{(L+1)}$ using an FFT of length 2^{L+1}.*
2. *Determine the first support index $\mu^{(L+1)} \in \{0, \ldots, 2^{L+1}-1\}$ of $\mathbf{x}^{(L+1)}$ such that $x^{(L+1)}_{\mu^{(L+1)}} \neq 0$ and $x^{(L+1)}_k = 0$ for $k \notin \{(\mu^{(L+1)} + r) \bmod 2^{L+1},\ r = 0, \ldots, m-1\}$.*
3. *Choose a Fourier component $\widehat{x}_{2k_0+1} \neq 0$ of $\widehat{\mathbf{x}}$ and compute the sum*
$$a := \sum_{\ell=0}^{m-1} x^{(L+1)}_{(\mu^{(L+1)}+\ell) \bmod 2^{L+1}}\, \omega_N^{(2k_0+1)(\mu^{(L+1)}+\ell)}.$$
4. *Compute $b := \widehat{x}_{2k_0+1}/a$ that is by construction of the form $b = \omega^p_{2^{J-L-1}}$ for some $p \in \{0, \ldots, 2^{J-L-1}-1\}$, and find $\nu \in \{0, \ldots, 2^{J-L-1}-1\}$ such that $(2k_0+1)\,\nu = p \bmod 2^{J-L-1}$.*
5. *Set $\mu^{(J)} := \mu^{(L+1)} + 2^{L+1}\nu$, and $\mathbf{x} := (x_k)_{k=0}^{N-1}$ with entries*
$$x_{(\mu^{(J)}+\ell) \bmod N} := \begin{cases} x^{(L+1)}_{(\mu^{(L+1)}+\ell) \bmod 2^{L+1}} & \ell = 0, \ldots, m-1, \\ 0 & \ell = m, \ldots, N-1. \end{cases}$$

Output: $\mathbf{x}$.

Our algorithm has an arithmetical complexity of $\mathcal{O}(m \log m)$. This can be seen as follows: In the first step, an FFT algorithm of this complexity is performed. All further steps require at most $\mathcal{O}(m)$ operations. Moreover, the algorithm needs less than $4m$ Fourier values.

The results can be found in a more detailed version in [5] where we also propose an algorithm for noisy input data as well as numerical results.

Acknowledgements We gratefully acknowledge the funding of this work by the DFG in the project PL 170/16-1.

References

[1] A. Gilbert, P. Indyk, M.A. Iwen, and L. Schmidt, IEEE Signal Processing Magazine **31**(5), pp. 91–100 (2014).
[2] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, Proc. 44th annual ACM symposium on Theory of Computing, pp. 563–578 (2012).
[3] D. Lawlor, Y. Wang, and A. Christlieb, Adv. Adapt. Data Anal. **5**(1), 1350003 (2013).
[4] S. Pawar and K. Ramchandran, IEEE International Symposium on Information Theory, pp. 464–468 (2013).
[5] G. Plonka, K. Wannenwetsch, arXiv:1504.02214 (2015).

