1. Using the Discrete Fourier Transform (DFT) to compute sums over Dirichlet Characters

In 3.2.9 of [1], a method of computing sums over Dirichlet Characters efficiently using the DFT. This note expands on that with some advice on how one might implement this in practice. In what follows, we adopt the 'C' convention that the first element of a vector X is $X[0]$ and that we can consider a vector X of length $l = \prod_{j=1}^{d} l_j$ as a d-dimensional vector $X[l_1, l_2, \ldots l_d]$.

Thus, we wish to populate a length $\phi(q)$ vector of complex numbers a_i in such a way that performing a (series of) DFT('s) on (portions of) that vector will compute

$$S_\chi = \sum_{i=1}^{q-1} \chi(i) a_i$$

for each of the $\phi(q)$ Dirichlet characters mod q. For example, to compute $L_\chi(1/2)$ one would take the a_i to be $\zeta(1/2, i/q)$.

1.1. Case 1.

Suppose q has a primitive root ω, in other words, $q \in \{4, p^n\}$ for p an odd prime and $n \geq 1$. Then we put a_i in the k'th position such that

$$\omega^k \equiv i \pmod{q}.$$

For example, let $q = 9$ so that our vector X consists of $\phi(9) = 6$ complex values, indexed [0..5]. One choice for the primitive root is $\omega = 2$ which means:

$$\begin{aligned}
2^0 &\equiv 1 \pmod 9, \text{so } a_1 \text{ goes in } X[0],\\
2^1 &\equiv 2 \pmod 9, \text{so } a_2 \text{ goes in } X[1],\\
2^2 &\equiv 4 \pmod 9, \text{so } a_4 \text{ goes in } X[2],\\
2^3 &\equiv 8 \pmod 9, \text{so } a_8 \text{ goes in } X[3],\\
2^4 &\equiv 7 \pmod 9, \text{so } a_7 \text{ goes in } X[4] \text{ and}\\
2^5 &\equiv 5 \pmod 9, \text{so } a_5 \text{ goes in } X[5].
\end{aligned}$$

We then do a length 6 DFT on X.

1.2. Case 2.

Now suppose $q = 2^n$ with $n \geq 3$ (so q does not have a primitive root). We treat our vector X as a $2 \times q/4$ matrix. We start with the first row using the pseudo primitive root 5 so $a_{5^k \pmod q}$ goes in $X[0, i]$, for $k = 0 \ldots q/4 - 1$. To populate the second row, we put $a_{-(5^k) \pmod q}$ in $X[1, i]$ for $k = 0 \ldots q/4 - 1$.

Another example, take $q = 16$, $\phi(q) = 8$. We get

$$
\begin{aligned}
5^0 &\equiv 1 \pmod{16}, \text{so } a_1 \text{ goes in } X[0,0],\\
5^1 &\equiv 5 \pmod{16}, \text{so } a_5 \text{ goes in } X[0,1],\\
5^2 &\equiv 9 \pmod{16}, \text{so } a_9 \text{ goes in } X[0,2],\\
5^3 &\equiv 13 \pmod{16}, \text{so } a_{13} \text{ goes in } X[0,3],\\
-(5^0) &\equiv 15 \pmod{16}, \text{so } a_{15} \text{ goes in } X[1,0],\\
-(5^1) &\equiv 11 \pmod{16}, \text{so } a_{11} \text{ goes in } X[1,1],\\
-(5^2) &\equiv 7 \pmod{16}, \text{so } a_7 \text{ goes in } X[1,2] \text{ and}\\
-(5^3) &\equiv 3 \pmod{16}, \text{so } a_3 \text{ goes in } X[1,3].
\end{aligned} \tag{1.1}
$$

We perform 2 length $q/4$ DFT's on the rows of X and $q/4$ length 2 DFT's on its columns. [1]

1.3. **Case 3.** Now our q is the product of more than one prime power with the power of 2 (if present) equal to 4.[2] Given d prime powers, we construct a d-dimensional matrix. If the factorisation is

$$q = \prod_{j=1}^{d} p_j^{\alpha_j}$$

then the marix will have dimensions

$$[\phi(p_1^{\alpha_1}), \phi(p_2^{\alpha_2}), \ldots \phi(p_d^{\alpha_d})].$$

We now do the obvious. Let ω_j be the selected primitive root for $p_j^{\alpha_j}$. Then the entry at $[c_1, c_2, \ldots, c_d]$ will be a_i such that

$$i \equiv \omega_j^{c_j} \pmod{p_j^{\alpha_j}} \text{ for } j \in 1 \ldots d.$$

As an example, consider $q = 45 = 3^3 \cdot 5$. We have $\phi(9) = 6$ and $\phi(5) = 4$ and we can use $\omega_1 = 2$ and $\omega_2 = 2$ as primitive roots. Our target matrix will have dimensions $[6, 4]$. Let us consider where to put a_{22}. We have

$$22 \equiv 4 \equiv 2^2 \pmod{9} \text{ and}$$
$$22 \equiv 2 \equiv 2^1 \pmod{5}$$

so a_{22} goes in $X[2, 1]$. Once we have populated all the $\phi(45) = 24$ locations, we do a length 4 DFT on each of the 6 rows and a length 6 DFT on each of the 4 columns.

1.4. **Case 4.** If we have a power of 2 (greater than 2^2) and d odd prime powers, we construct a $d + 2$-dimensional matrix. If we have

$$q = 2^{\alpha} \prod_{j=1}^{d} p_j^{\alpha_j}$$

then the dimensions of the matrix will be

$$[2, 2^{\alpha-2}, \phi(p_1^{\alpha_1}), \phi(p_2^{\alpha_2}), \ldots \phi(p_d^{\alpha_d})].$$

[1] A length 2 DFT simply replaces the two entries with their sum and difference.

[2] We do not consider $q \equiv 2 \pmod 4$ as there are no primitive Dirichlet characters to worry about.

We proceed exactly as in Case 3 above, except that when considering where to put a_i, if $i \equiv 5^k \pmod{2^\alpha}$ then a_i goes in $[0, k, \ldots]$ and if we have $i \equiv -(5^k) \pmod{2^\alpha}$ then a_i goes in $[1, k, \ldots]$.

As an example here, consider $q = 72 = 2^3 \cdot 3^2$. We construct X to have dimensions $[2, 2, 6]$. We can now determine where to put a_{29} by noting that

$$29 \equiv 5 \equiv 5^1 \pmod{8} \text{ and}$$
$$29 \equiv 2 \equiv 2^1 \pmod{9}$$

so a_{29} goes in $X[0, 1, 1]$. We then do 12 length 2 DFT's along dimension 1, 12 length 2 DFT's along dimension 2 and 4 length 6 DFT's along dimension 3.

1.5. **A note on complexity.** There are implementations of the DFT that achieve time complexity $\mathcal{O}(n \log n)$, even when the length n is prime. These are collectively known as Fast Fourier Transforms.

Thus in Case 1, a length $\phi(q)$ DFT results in computing $\phi(q)$ sums in time $\mathcal{O}(\phi(q) \log \phi(q))$, or $\mathcal{O}(q^\epsilon)$ on average per value. It turns out we achieve the same time complexity in the other two cases as well.

1.6. **A note on Primitive Characters.** At least one of the characters mod q will be imprimitive. It is helpful to know where to find the primitive results in the multi-dimensional matrix that results from the above. Fortunately, it is straight forward. We have two cases:-

1.6.1. $8 \nmid q$. Let the dimensions of the matrix be $[l_1, l_2, \ldots l_d]$ corresponding to prime powers of $[p_1, p_2, \ldots p_d]$ then any entry $[c_1, c_2, \ldots c_d]$ with any $c_i \equiv 0 \pmod{p_i}$ is imprimitive.

1.6.2. $8|q$. Let the dimensions of the matrix be $[2, 2^{\alpha-2}, l_1, l_2, \ldots l_d]$ corresponding to prime powers of $p_1, p_2, \ldots p_d]$ then any entry $[*, c, c_1, c_2, \ldots c_d]$ with $c \equiv 0 \pmod{2}$ or any $c_i \equiv 0 \pmod{p_i}$ is imprimitive.

1.7. **A note on the parity of characters.** We will also want to be able to locate where the odd ($\chi(-1) = -1$) and even ($\chi(-1) = 1$) characters are hiding in our data structure. It turns out (proof anyone?) that if one sums the co-ordinates relating to the character of interest, that character is odd iff the sum is odd.

Taking our ??????????

1.8. **A suitable FFT Algorithm.** The most well known FFT algorithm (after Euler) requires the input vector be be of length 2^n for positive n. Most of the time, the DFT's we require will not be of such a convenient length. We employ a FFT known as Bluestein's algorithm or the Chirp-Z transform algorithm converts a DFT of arbitrary length n into the circular convolution of two inputs of length n and $2n - 1$. As is well know, a circular convolution can be achieved via three DFT's. The advantage is the one **can** zero pad the input vectors for a circular convolution so we can engineer it so that all the DFT's are of a convenient length ($2^{n'}$ with $2^{n'} > 2n - 1$).

A good online description of Bluestein can be found at `https://ccrma.stanford.edu/~jos/st/Bluestein_s_FFT_Algorithm.html`.

References

[1] David J. Platt. *Computing Degree 1 L-functions Rigorously.* PhD thesis, University of Bristol, 2011.