Fast Fourier Transform

Algorithm: fast multiplication of polynomials

The following algorithm given by its high-level description can be used for fast multiplication of two polynomials, given by their coefficients.

**Input:** Two polynomials \( A(x) = \sum_{i=0}^{m} a_i x^i \) and \( B(x) = \sum_{i=0}^{m} b_i x^i \) with real coefficients.

**Output:** Polynomial \( C(x) = \sum_{i=0}^{n-1} c_i x^i \), such that \( C(x) = A(x) \cdot B(x) \), \( n \geq 2m + 1 \).

1. **Point selection.** Select \( n \) distinct points \( x_0, x_1, \ldots, x_{n-1} \), \( n \geq 2m + 1 \).

2. **Evaluation.** Compute \( A(x_0), A(x_1), \ldots, A(x_{n-1}) \) and \( B(x_0), B(x_1), \ldots, B(x_{n-1}) \).

3. **Multiplication.** For \( i = 0, 1, \ldots, n - 1 \), compute \( C(x_i) = A(x_i) \cdot B(x_i) \).

4. **Interpolation.** Recover \( C(x) = \sum_{i=0}^{n-1} c_i x^i \).

Reminder: complex numbers

A complex number \( z = a + i \cdot b \), where \( i = \sqrt{-1} \), can be viewed as a point \((a, b)\) on the coordinate plane. This representation is called a representation in Cartesian coordinates. The representation in polar coordinates is \((r, \theta)\), where \( r = \sqrt{a^2 + b^2} \) and \( \theta = \arctan \left( \frac{b}{a} \right) \in [0, 2\pi) \). Given a representation in polar coordinates of a complex number \( z = (r, \theta) \), the equivalent representation in Cartesian coordinates is \((a, b)\), where \( a = r \cos \theta \) and \( b = r \sin \theta \).

Assume that we have two complex numbers in their polar coordinate representation, \( z_1 = (r_1, \theta_1) \) and \( z_2 = (r_2, \theta_2) \). Then their product \( z = z_1 \cdot z_2 \) satisfies \( z = (r_1 \cdot r_2, \theta_1 + \theta_2) \).

**Definition 1** For any positive integer \( n \), the \( n \)-th root of unity is a solution \( z \) to the equation \( z^n = 1 \).

The \( n \)-th roots of unity in their polar coordinate representations are

\[(1, 0), \left(1, \frac{2\pi}{n}\right), \left(1, 2 \cdot \frac{2\pi}{n}\right), \left(1, 3 \cdot \frac{2\pi}{n}\right), \ldots, \left(1, (n-1) \cdot \frac{2\pi}{n}\right) \, .\]

There are \( n \) such roots of unity. The root \( \omega = (1, \frac{2\pi}{n}) \) is called primitive. Then, all \( n \) roots of unity are obtained as powers of \( \omega \), namely \( 1, \omega, \omega^2, \omega^3, \ldots, \omega^{n-1} \).

---

Matrix representation

If \( A(x) = \sum_{i=0}^{n-1} a_i x^i \), then the following equation holds:

\[
\begin{pmatrix}
A(x_0) \\
A(x_1) \\
A(x_2) \\
\vdots \\
A(x_{n-1})
\end{pmatrix}
= \begin{pmatrix}
1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\
1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\
1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1}
\end{pmatrix}
\begin{pmatrix}
a_0 \\
a_1 \\
a_2 \\
\vdots \\
a_{n-1}
\end{pmatrix}.
\]

The matrix above belongs to the so-called class of Vandermonde matrices. We choose \( x_0, x_1, \ldots, x_{n-1} \) to be \( n \) distinct \( n \)-th roots of unity. For this selection of points, the matrix above becomes

\[
M_n(\omega) \triangleq \begin{pmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\
1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)}
\end{pmatrix}.
\]

The matrix \( M_n(\omega) \) is symmetric. Multiplication of a vector by such matrix is called Fast Fourier Transform.

The following theorem tells us that the inverse of the matrix \( M_n(\omega) \) has a form very similar to the form of the matrix \( M_n(\omega) \).

**Theorem 1** We have

\[
(M_n(\omega))^{-1} = \frac{1}{n} M_n(\omega^{-1}).
\]

It follows from the theorem that interpolation in Step 4 of the algorithm is equivalent to multiplication by \( M_n(\omega^{-1}) \), and is itself Fast Fourier Transform operation.

**Proof.** We compute the product \( M_n(\omega) \cdot M_n(\omega^{-1}) \). The elements on the main diagonal satisfy (for all \( i = 0, 1, \ldots, n-1 \)):

\[
\left( M_n(\omega) \cdot M_n(\omega^{-1}) \right)_{i,i} = \sum_{k=0}^{n-1} \omega^{ik} \cdot \omega^{-ki} = \sum_{k=0}^{n-1} 1 = n.
\]

For all \( i = 0, 1, \ldots, n-1 \) and \( j = 0, 1, \ldots, n-1 \), such that \( i \neq j \), we obtain

\[
\left( M_n(\omega) \cdot M_n(\omega^{-1}) \right)_{i,j} = 1 \cdot 1 + \omega^i \cdot \omega^{-j} + \omega^{2i} \cdot \omega^{-2j} + \cdots + \omega^{i(n-1)} \cdot \omega^{-(n-1)j} = \sum_{k=0}^{n-1} \omega^{k(i-j)} = \frac{1 - \omega^{n(i-j)}}{1 - \omega^{i-j}},
\]

(1)
where the last transition is obtained by summing up the elements in the geometric series.

Here, $1 - \omega^{i-j} \neq 0$ and $1 - \omega^{n(i-j)} = 0$. Therefore, the expression in (1) is equal to zero. □

The element in row $j$ and column $k$ of $M_n(\omega)$ is $\omega^{jk}$, for $j = 0, 1, \cdots, n - 1$, and for $k = 0, 1, \cdots, n - 1$. Now, consider the elements in rows $j$ and $j + n/2$ for $j = 0, 1, \cdots, n/2 - 1$, and in columns $2k$ and $2k + 1$ for $k = 0, 1, \cdots, n/2 - 1$. Their values are given in the following table.

<table>
<thead>
<tr>
<th>Row $j$</th>
<th>Column $2k$</th>
<th>Column $2k + 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j$</td>
<td>$\omega^{2jk}$</td>
<td>$\omega^{j+2jk}$</td>
</tr>
<tr>
<td>$j + n/2$</td>
<td>$\omega^{2jk}$</td>
<td>$-\omega^{j+2jk}$</td>
</tr>
</tbody>
</table>

We have that

$$
\begin{pmatrix}
A(1) \\
A(\omega^1) \\
A(\omega^2) \\
\vdots \\
A(\omega^{n-1})
\end{pmatrix} = \begin{pmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\
1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)}
\end{pmatrix} \cdot \begin{pmatrix}
a_0 \\
a_1 \\
a_2 \\
\vdots \\
a_{n-1}
\end{pmatrix}.
$$

Schematically this product can be presented as follows:

Row $j$ → $M_{n/2}(\omega^2) \cdot \begin{pmatrix}
a_0 \\
a_2 \\
\vdots \\
a_{n-2}
\end{pmatrix} + \omega^j \cdot M_{n/2}(\omega^2) \cdot \begin{pmatrix}
a_1 \\
a_3 \\
\vdots \\
a_{n-1}
\end{pmatrix}$

Row $j + n/2$ → $M_{n/2}(\omega^2) \cdot \begin{pmatrix}
a_0 \\
a_2 \\
\vdots \\
a_{n-2}
\end{pmatrix} - \omega^j \cdot M_{n/2}(\omega^2) \cdot \begin{pmatrix}
a_1 \\
a_3 \\
\vdots \\
a_{n-1}
\end{pmatrix}$

We observe that the multiplication of $n \times n$ matrix $M_n(\omega)$ by a length-$n$ vector of coefficients reduces to two multiplications of $(n/2) \times (n/2)$ matrix $M_{n/2}(\omega^2)$ by two length-$n/2$ vectors of coefficients, plus $O(n)$ additional operations.

**Complexity analysis**

Let $n$ be a power of 2, and let $T(n)$ be time required to substitute $n$ roots of unity into polynomial of degree $\leq n - 1$. Then, the complexity of this operation is given by

$$
T(n) = 2 \cdot T(n/2) + O(n) = \cdots = O(n \log n).
$$

The algorithm consists of two evaluations of polynomials in Step 2, $O(n \log n)$ each, of $n$ multiplications in Step 3, and of interpolation in Step 4, which requires additional $O(n \log n)$ time. The total complexity is, therefore, $O(n \log n)$.  

3