



БЫСТРЫЕ АЛГОРИТМЫ ДИСКРЕТНОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ

Алмазова Гулшат Мырадовна

Старший преподаватель Туркменский государственный архитектурно-строительный институт г. Ашхабад Туркменистан

Рахманбердийев Шамухаммет Баймухаммедович

Преподаватель Туркменский государственный архитектурно-строительный институт г. Ашхабад Туркменистан

В работе рассматриваются быстрые алгоритмы дискретного преобразования Фурье. Данные алгоритмы позволяют снизить вычислительную сложность обычного дискретного преобразования Фурье.

Введение

Пусть $\mathbf{v} = \{v_i, i = 0, \dots, n-1\}$ обозначает вектор с вещественными или комплексными компонентами.

Определение. Преобразованием Фурье вектора $\mathbf{v}$ называется вектор $\mathbf{V}$ длины n с комплексными компонентами, задаваемыми равенствами

$$V_k = \sum_{i=0}^{n-1} \omega^{ik} v_i, k = 0, \dots, n-1,$$

где: $\omega = e^{-\frac{j2\pi}{n}}$.

Преобразование Фурье, записанное в таком виде требует n^2 сложений и n^2 умножений. Для уменьшения количества арифметических операций рассмотрим быстрые алгоритмы преобразования Фурье.

Алгоритм Кули-Тьюки в общем виде

Для построения алгоритма предполагается, что $n = n_1 n_2$. В выражении для преобразования Фурье произведем следующую замену записи индексов:

$$\begin{array}{lll} i = i_1 + n_1 i_2 & i_1 = 0, \dots, n_1-1 & i_2 = 0, \dots, n_2-1 \\ k = n_2 k_1 + k_2 & k_1 = 0, \dots, n_1-1 & k_2 = 0, \dots, n_2-1 \end{array}$$

Подставим новые обозначения в исходную формулу преобразования:

$$V_{n_2 k_1 + k_2} = \sum_{i_2=0}^{n_2-1} \sum_{i_1=0}^{n_1-1} \omega^{(i_1+n_1 i_2)(n_2 k_1 + k_2)} v_{i_1+n_1 i_2}$$

Определим двумерные переменные для удобства записи формул:

$$v_{i_1, i_2} = v_{i_1 + n_1 i_2}, V_{k_1, k_2} = V_{n_2 k_1 + k_2}.$$

Раскроем скобки в показателе степени и положим $\omega^{n_1} = \gamma$ и $\omega^{n_2} = \beta$. Также примем во внимание, что $\omega^{n_1 n_2 i_2 k_1} = \omega^{n i_2 k_1} = 1$.

В терминах двумерных переменных формула преобразуется к виду

$$V_{k_1, k_2} = \sum_{i_1=0}^{n_1-1} \beta^{i_1 k_1} \left[\omega^{i_1 k_2} \sum_{i_2=0}^{n_2-1} \gamma^{i_2 k_2} v_{i_1, i_2} \right]$$

Дискретное преобразование Фурье по алгоритму Кули-Тьюки требует не более $n^*(n_1 + n_2 + 1)$ комплексных умножений и $n^*(n_1 + n_2 - 2)$ комплексных сложений. В общем случае количество умножений и сложений можно записать следующим образом:

$$M_C(n) = n_1 M_C(n_2) + n_2 M_C(n_1) + n$$

$$A_C(n) = n_1 A_C(n_2) + n_2 A_C(n_1)$$

Так, например, для $n = 1000$ (учитывая, что $M_C(2) = 0$, $M_C(5) = 10$):

$$M_C(1000) = 2M_C(500) + 500M_C(2) + 1000 = 2M_C(500) + 1000 = 11000$$

Таким образом, алгоритм Кули-Тьюки дает выигрыш по количеству умножений в $1000000/11000 \approx 91$ раз.

Визуально алгоритм выглядит следующим образом: значения вектора $\mathbf{v}$ заносятся по порядку в таблицу n_1 на n_2 слева направо, сверху вниз. Вычисления состоят из n_2 -точечного ДПФ каждого столбца, поэлементного умножения всех элементов таблицы соответственно на $\omega^{i_1 k_2}$ и n_1 -точечного ДПФ для каждой строки.

Алгоритм Кули-Тьюки в случае, когда $n = 2^m$

Если $n_1 = 2$, а $n_2 = 2^{m-1}$, то алгоритм называют алгоритмом Кули-Тьюки по основанию два с *прореживанием по времени*. Используя тот факт, что $\beta = \omega^{n/2} = -1$, уравнения, задающие БПФ, можно записать в следующем виде:

$$V_k = \sum_{i=0}^{\frac{n}{2}-1} \omega^{2ik} v_{2i} + \omega^k \sum_{i=0}^{\frac{n}{2}-1} \omega^{2ik} v_{2i+1}$$

$$V_{k+\frac{n}{2}} = \sum_{i=0}^{\frac{n}{2}-1} \omega^{2ik} v_{2i} - \omega^k \sum_{i=0}^{\frac{n}{2}-1} \omega^{2ik} v_{2i+1}$$

$$k = 0, \dots, n/2 - 1$$

Если $n_1 = 2^{m-1}$, а $n_2 = 2$, то алгоритм называют алгоритмом Кули-Тьюки по основанию два с *прореживанием по частоте*. Уравнения БПФ в этом случае преобразуются к виду:

$$V_{2k_1} = \sum_{i_1=0}^{\frac{n}{2}-1} \left(v_{i_1} + v_{i_1+\frac{n}{2}} \right) \omega^{2i_1k_1}$$

$$V_{2k_1+1} = \sum_{i_1=0}^{\frac{n}{2}-1} \left(v_{i_1} - v_{i_1+\frac{n}{2}} \right) \omega^{2i_1k_1} \omega^{i_1}$$

Число $M_C(n)$ комплексных умножений n -точечного БПФ удовлетворяет уравнению $M_C(n) = 2M_C(n/2) + n/2$, а число $A_C(n)$ комплексных сложений удовлетворяет уравнению $A_C(n) = 2A_C(n/2) + n$. Решения этих уравнений даются равенствами

$$M_C(n) = (n/2) \log_2 n,$$

$$A_C(n) = n \log_2 n$$

Алгоритм Гуда-Томаса

Для применения алгоритма Гуда-Томаса числа n_1 и n_2 должны быть взаимно простыми. В двумерную таблицу входные данные выписываются, начиная с верхнего левого угла, вдоль расширенной диагонали. Входные индексы задаются по правилу

$$i_1 = i \pmod{n_1}, \quad i_2 = i \pmod{n_2}$$

Это правило представляет собой отображение индекса i на расширенную диагональ двумерной таблицы, элементы которой занумерованы парами индексов (i_1, i_2) . Из теории чисел известно, что для чисел n_1 и n_2 существуют такие целые числа N_1 и N_2 , что $N_1 n_1 + N_2 n_2 = \text{НОД}(n_1, n_2)$. Если n_1 и n_2 взаимно просты, то, согласно китайской теореме об остатках

$$i = i_1 N_2 n_2 + i_2 N_1 n_1 \pmod{n}.$$

Пусть $k_1 = N_2 k \pmod{n_1}$, $k_2 = N_1 k \pmod{n_2}$. Тогда выходные индексы вычисляются по правилу $k = n_2 k_1 + n_1 k_2 \pmod{n}$.

В этих новых индексных обозначениях формула преобразуется к виду

$$V_{n_2 k_1 + n_1 k_2} = \sum_{i_2=0}^{n_2-1} \sum_{i_1=0}^{n_1-1} \omega^{(i_1 N_2 n_2 + i_2 N_1 n_1)(n_2 k_1 + n_1 k_2)} v_{i_1 N_2 n_2 + i_2 N_1 n_1}$$

Раскроем скобки в показателе степени

$$V_{k_1, k_2} = \sum_{i_1=0}^{n_1-1} \sum_{i_2=0}^{n_2-1} \omega^{N_2(n_2)^2 i_1 k_1} \omega^{N_1(n_1)^2 i_2 k_2} v_{i_1, i_2} = \sum_{i_1=0}^{n_1-1} \sum_{i_2=0}^{n_2-1} \beta^{i_1 k_1} \gamma^{i_2 k_2} v_{i_1, i_2},$$

где: $\beta = \omega^{N_2(n_2)^2} = e^{-j2\pi/n_1}$, $\gamma = \omega^{N_1(n_1)^2} = e^{-j2\pi/n_2}$.

Если длина n преобразования раскладывается в произведение простых множителей n_l , то описанная форма БПФ-алгоритма требует примерно $n \sum n_l$ умножений и столько же сложений. Например, для $n = 1000 = 2*2*2*5*5*5$ число умножений $MC(1000) = 1000*(2+2+2+5+5+5) = 21000$, что дает выигрыш по количеству операций умножения в $1000000/21000 \approx 48$ раз.

Заключение

В работе были рассмотрены два наиболее известных алгоритма дискретного преобразования Фурье, а также рассчитаны выигрыши по количеству умножений данных алгоритмов по сравнению с обычным алгоритмом преобразования Фурье.