Hadamard matrix times vector in \(O(n \log n)\)

Manfred Warmuth

The Hadamard matrices \(H_0, H_1, H_2, \ldots\) are defined as follows:

- \(H_0\) is the \(1 \times 1\) matrix \([1]\)
- For \(k > 0\), \(H_k\) is the \(2^k \times 2^k\) matrix

\[
H_k = \begin{bmatrix}
H_{k-1} & H_{k-1} \\
H_{k-1} & -H_{k-1}
\end{bmatrix}
\]

Show that if \(v\) is a column vector of length \(n = 2^k\), then the matrix-vector product \(H_k v\) can be calculated using \(O(n \log n)\) operations instead of \(O(n^2)\). Assume that all the numbers involved are small enough that basic arithmetic operations like addition and multiplication take unit time.

Solution:

Let \(v\) be an arbitrary column vector of size \(n = 2^k\). Also let \(v_u\) and \(v_l\) be two column vectors of size \(n/2 = 2^{k-1}\) representing the upper and lower halves of \(v\) i.e. \(v = \begin{bmatrix} v_u \\ v_l \end{bmatrix}\). Therefore, we have:

\[
H_k \cdot v = \begin{bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{bmatrix} \begin{bmatrix} v_u \\ v_l \end{bmatrix} = \begin{bmatrix} H_{k-1} \cdot v_u + H_{k-1} \cdot v_l \\ H_{k-1} \cdot v_u - H_{k-1} \cdot v_l \end{bmatrix}
\]

Thus in order to find \(H_k \cdot v\) it suffices to find \(H_{k-1} \cdot v_u\) and \(H_{k-1} \cdot v_l\) and then compute \(H_{k-1} \cdot v_u \pm H_{k-1} \cdot v_l\). See Algorithm 1.

**Algorithm 1** \(O(n \log n)\) Algorithm for Finding Hadamard Matrix-Vector Multiplication

```plaintext
procedure HADAMARD(k, v)
    if k = 0 then
        return v - \{ v is a scalar in this case \}
    end if
    v_l ← LOWER(v)
    v_u ← UPPER(v)
    m_l ← HADAMARD(k - 1, v_l)
    m_u ← HADAMARD(k - 1, v_u)
    m ← \[m_u + m_l\] - \{ Takes \(O(n)\) time for elementwise vector addition and subtraction\}
    return m
end procedure
```

If \(T(n)\) is the cost for finding \(H_k \cdot v\), then we can find each \(H_{k-1} \cdot v_u\) and \(H_{k-1} \cdot v_l\) in \(T(n/2)\). In addition, \(H_{k-1} \cdot v_u \pm H_{k-1} \cdot v_l\) can be found in linear time since we need to do elementwise vector addition and subtraction. Therefore:

\[
T(n) = 2 \cdot T(n/2) + O(n)
\]

Observe that \(f(n) = O(n)\) and \(1 = \log_2 2\). Thus using Master Theorem, we obtain \(T(n) = \Theta(n \log n)\).