FWT and Bent Functions

Alberto Ibrisevic
Paolo Piasenti

July 2021

Il seguente elaborato descrive l'algoritmo noto come Fast Walsh Transform (abbreviato FWT), che permette di calcolare in modo efficiente la trasformata di Walsh per una funzione booleana $f : (\mathbb{F}_2)^n \to \mathbb{F}_2$. Verranno innanzitutto introdotte le nozioni teoriche fondamentali dietro l'algoritmo stesso e le loro relazioni con i grafi di Cayley associati a f.

1 Nozioni preliminari

Prima di descrivere l'algoritmo della FWT, introduciamo le definizioni fondamentali e alcuni principali risultati . Una funzione $f : (\mathbb{F}_2)^n \to \mathbb{F}_2$ viene detta *funzione booleana* in n variabili. Si definisce la *truth table* di f come il vettore $v = (f(b(0)), \dots, f(b(2^n - 1))) \in (\mathbb{F}_2)^{2^n}$, dove $b_n : \{0, \dots, 2^n - 1\} \to \mathbb{F}_2^n$ è la rappresentazione binaria degli interi in $[0, 2^n - 1]$. Si definisce inoltre il *supporto* di f come l'insieme $\Omega_f := \{i \in \{0, \dots, 2^n - 1\} : v_i = 1\} \subseteq \{0, \dots, 2^n - 1\}$, mentre il *peso* di f (ossia l'*Hamming Weight*) è dato dalla cardinalità di Ω_f, i.e. $|f| := |\Omega_f|$. In generale ogni funzione booleana in n variabili può essere rappresentata mediante un polinomio $p(x_1, \dots, x_n) \in \mathbb{F}_2[x_1, \dots, x_n]$, in cui ogni monomio è square-free (poiché essendo in caratteristica 2, si ha naturalmente che $x^2 \equiv x \mod 2$; tale rappresentazione mediante il polinomio p viene detta *Absolute Normal Form (ANF)*. Si ha dunque il seguente risultato: data una funzione booleana $f : \mathbb{F}_2^n \to \mathbb{F}_2$, è sempre possibile rappresentare la sua ANF, più precisamente

$$f(x) = \sum_{a \in \mathbb{F}_2^n} c_a x_1^{a_1} \cdots x_n^{a_n},$$

con $a = (a_1, \dots, a_n)$ e $c_a = \sum_{x \leq a} f(x)$ (dove per "$x \leq a$" si intende "$x_i \leq a_i \quad \forall 1 \leq i \leq n$").

Definiamo ora la *Walsh transform* di una funzione booleana f come la mappa $W_f : \mathbb{F}_2^n \to \mathbb{R}$ tale che

$$W_f(a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{a \cdot x \oplus f(x)}.$$

Il vettore $w = (W_f(b(0), \ldots, W_f(b(2^n - 1))) \in (\mathbb{F}_2)^{2^n}$ è chiamato *spettro* della Walsh transform di f. L'applicazione della trasformata W_f può essere anche riscritta in forma matriciale nel seguente modo,

$$w = H_n \cdot \overline{v},$$

dove $\overline{v}$ è la *polarity truth table* di f, definita come $\overline{v} = (1, \ldots, 1) - 2 \cdot (v_1, \ldots, v_n)$, e H_n è una matrice $2^n \times 2^n$, detta *matrice di Hadamard*, le cui entrate sono $h_{ij} = \pm 1$. Più precisamente, si possono costruire matrici di Hadamard in modo ricorsivo nel seguente modo:

$$H_0 = \begin{pmatrix} 1 \end{pmatrix}, \qquad H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$
$$H_{n+1} = \begin{pmatrix} H_n & H_n \\ H_n & -H_n \end{pmatrix}$$

Si può osservare che la trasformata di Walsh non è che un caso particolare di trasformata discreta di Fourier nel gruppo abeliano finito $G = \mathbb{F}_2^n$. Infatti, prendendo ad esempio $n = 1$, quindi $G = \mathbb{F}_2$, il suo gruppo duale è $\hat{G} = \{e_0, e_1\} = \{(1, 1), (1, -1)\}$, ossia $F = H_1$. Una proprietà molto importante che riguarda lo spettro della Walsh trnasform è la seguente.

Proposizione. *(Equazione di Parseval) Per ogni funzione booleana f in n variabili, vale la seguente equazione*

$$\sum_{u \in \mathbb{F}_2^n} \big(W_f(u)\big)^2 = 2^{2n}$$

2 The Fast Walsh Transform

Data la struttura delle matrici di Hadamard H_n, è possibile eseguire un algoritmo ricorsivo che calcola la walsh transform di una funzione booleana $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ in n variabili a partire da quella della funzione f' in $n-1$ variabili attraverso il criterio del "dividi-et-impera". A livello matriciale la questione si traduce dal calcolare $w = H_n f^t$, dove H_n ha ordine $m = 2^n$, a partire invece dalla matrice H_{n-1} di ordine $2^{n-1} = m/2$. A livello computazionale, anziché eseguire $O(m^2)$ operazioni, l'algoritmo permette di ottenere la trasformata di f in $O(m \log m)$ addizioni e sottrazioni. L'idea dell'algoritmo è quella di riscrivere l'equazione nella forma

$$w = H_n f^t = \begin{pmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{pmatrix} H_{n-1} f_1 + H_{n-1} f_2 \\ H_{n-1} f_1 - H_{n-1} f_2 \end{pmatrix},$$

dove $f_1 = (f(0), \ldots, f(2^{n-1} - 1))^t$ e $f_2 = (f(2^{n-1}), \ldots, f(2^n - 1))^t$. Al primo passo dell'iterazione quindi si computano gli spettri $H_{n-1} f_1 = w_1$ e $H_{n-1} f_2 = w_2$, utilizzando solo un quarto della matrice di partenza, e lo spettro w sarà dato dalla concatenazione dei vettori $w_1 + w_2$ e $w_1 - w_2$. Si itera poi questo

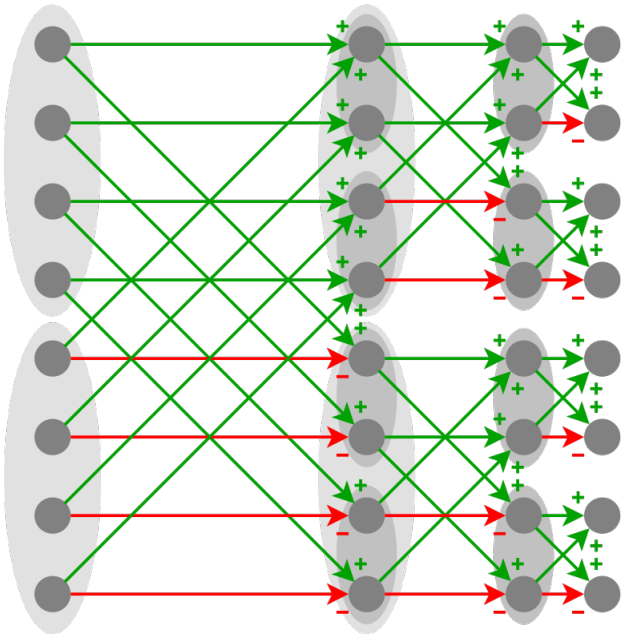

Figura 1: Fast Walsh Transform applicata ad una funzione in 3 variabili (e dunque 8 valutazioni possibili) [fonte: Wikipedia]

procedimento per n volte, fino ad arrivare al caso base con $H_0 = (1)$, utilizzando solo somme o sottrazioni da un passo al successivo. Lo pseudocodice risultante è dunque riportato in 1.

Algorithm 1 Fast Walsh Transform

```
1:  function FWT(PTT)
2:  Input: La polarity truth table PTT della funzione f
3:  Output: Lo spettro della Walsh transform W_f
4:      size ← 1
5:      W_f ← PTT
6:      while size < 2^n do
7:          for pos = size − 1 to 2^n by 2 · size do
8:              for j = pos + 1 to 2pos + size + 1 do
9:                  a ← W_f[j − size]
10:                 b ← W_f[j]
11:                 W_f[j − size] = a + b
12:                 W_f[j] = a − b
13:         size ← 2 · size
14:     return W_f
```

3 Nonlinearità e funzioni Bent

Procedendo con le nozioni teoriche, definiamo una la classe delle *funzioni affini* nell'insieme delle funzioni booleane in n variabili B_n come

$$A_n = \{\varphi_{a,c} \in B_n : \varphi_{a,c} = a \cdot x + c \mid a, c \in \mathbb{F}_2\}$$

Data una funzione booleana qualsiasi $f \in B_n$, si definisce la sua *nonlinearità* $N(f)$ come la minore tra le distanze rispetto ad ogni funzione affine $\phi \in A_n$, ossia

$$N(f) = \min_{\varphi \in A_n} d(f, \varphi),$$

dove $d(f, \varphi) = |f \oplus \varphi|$ è la distanza di Hamming. La nonlinearità di una funzione booleana è un parametro importante in ambito crittografico, più è grande e maggiore è la resistenza ad attacchi lineari (e.g. Berlekamp-Massey). Tuttavia, $N(f)$ è limitata dall'alto dalla seguente disuguaglianza:

$$N(f) \leq 2^{n-1} - 2^{n/2-1}$$

Per n pari, tale limite può essere raggiunto, e le funzioni booleane aventi questa caratteristica sono chimate funzioni *bent*. Un ulteriore risultato che unisce la nonlinearità di una funzione $f \in B_n$ allo spettro della Walsh transform W_f è dato dal seguente teorema.

Teorema 1. *La nonlinearità di f è determinata dalla Walsh transform W_f attraverso relazione*

$$N(f) = 2^{n-1} - \frac{1}{2} \max_{u \in \mathbb{F}_2^n} |W_f(u)|.$$

In particolare si avrà che se f è una funzione bent allora $\max_{u \in \mathbb{F}_2^n} |W_f(u)| = 2^{n/2}$, ossia $W_f(u) = \pm 2^{n/2}$.

Teorema 2. *Le seguenti affermazioni sono equivalenti:*

i. $f \in B_n$ *è bent, i.e.* $N(f) = 2^{n-1} - 2^{n/2-1}$

ii. $W_f(u) = \pm 2^{n/2}\ \forall u \in \mathbb{F}_2^n$

Dimostrazione.
i. $\implies$ *ii.* Supponendo che f sia una funzione bent, dall'osservazione fatta per il teorema 1 si ottiene immediatamente che $W_f(u) = |2^{n/2}|\ \forall u \in \mathbb{F}_2^n$ e quindi la tesi.
ii. $\implies$ *i.* Supponiamo per assurdo che esista $u \in \mathbb{F}_2^n$ tale che $W_f(u) \neq \pm 2^{n/2}$: se $|W_f(u)| > 2^{n/2}$, sempre utilizzando il teorema 1, si arriva all'assurdo $N(f) < 2^{n-1} - 2^{n/2-1}$; viceversa, se $|W_f(u)| < 2^{n/2}$, esisterà allora un altro elemento $z \in \mathbb{F}_2^n$ tale che $|W_f(u)| > 2^{n/2}$, altrimenti l'equazione di Parseval non sarebbe più soddisfatta, arrivando dunque anche qui all'assurdo. Pertanto lo spettro W_f è costante in valore assoluto e $|W_f(u)| = 2^{n/2}\ \forall u \in \mathbb{F}_2^n$. □

4 Grafo di Cayley di funzioni bent

In questa sezione siamo interessati a studiare il grafo di Cayley associato a una funzione booleana, ed in particolare se questa è una funzione bent.

Definizione 1. Sia $f \in B_n$ una funzione booleana in n variabili e Ω_f il suo supporto. Il grafo di Cayley associato a f è

$$G = X(\mathbb{F}_2^n, \Omega_f),$$

dove l'insieme dei vertici è $V(G) = \mathbb{F}_2^n$ e l'insieme degli archi è dato da $E(G) = \{\{v, w\} \mid v, w \in \mathbb{F}_2^n, f(v \oplus w) = 1\}$.

Questa è una buona definizione, infatti Ω_f è un insieme simmetrico (poiché $u = -u\ \forall u \in \mathbb{F}_2^n$) e se $f(v \oplus w) = 1$ allora $w = v \oplus s, s \in \omega_f$ (poiché $f(v \oplus w) = f(v \oplus v \oplus s) = f(s) = 1$ per definizione di Ω_f), pertanto $E(G) = \Omega_f$. Osserviamo inoltre che se $0 = (0, \ldots, 0) \notin \Omega_f$ allora G è semplice (i.e. non ha loop in nessun nodo). Infine il grafo così costruito è sempre $|\Omega_f|$-regolare. Lo spettro (degli autovalori λ) del grafo G_f è composto dai valori dalla Discrete Forier Transform di δ_{Ω_f}, ossia $Spec(G_f) = \{\hat{\delta}_{\Omega_F}(u) : u \in \mathbb{F}_2^n\} = \{\langle \delta_{\Omega_F}, \chi_u \rangle : u \in \mathbb{F}_2^n\}$. Dal momento che il gruppo su cui si sta calcolando la DFT è il gruppo abeliano

finito $\mathbb{F}_2^n$ i caratteri χ_u si ottengono facilmente dalle colonne della matrice di Hadamard H_n, pertanto gli autovalori λ_i sono dati da

$$\lambda_i = \sum_{x \in \mathbb{F}_2^n} (-1)^{x \cdot b(i)} f(x) = \hat{f}(b(i))$$

Ricordiamo le definizioni di grafi regolari e fortemente regolari. Un grafo G si dice *regolare* se il grado di ogni nodo v (i.e., il numero di archi uscenti da v) è costante. Un grafo *fortemente regolare* è un grafo regolare con l'ulteriore proprietà che per ogni coppia di nodi adiacenti ci sono esattamente e nodi vicini in comune, oppure f nodi altrimenti. Tali tipi di grafi presentano alcune peculiarità:

- Se G è un grafo k-regolare allora $\lambda_0 = k \in Spec(G)$
- G è un grafo fortemente regolare $\iff$ $Spec(G)$ ha esattamente tre autovalori distinti

Pertanto nel caso di $G = X(\mathbb{F}_2^n, \Omega_f)$ si ha che G è $|\Omega_f|$-regolare per ogni funzione booleana $f \in B_n$. Il seguente teorema invece mette in relazione le funzioni bent con una classe particolare di grafi fortemente regolari.

Teorema 3. *Sia $n > 2$. Allora la funzione $f \in B_n$ è bent se e solo se il relativo grafo di Cayley G_f è fortemente regolare con $e = f$.*

Sempre nelle ipotesi del teorema, si ha inoltre una relazione tra lo spettro del grafo G_f e quello della Walsh transform della funzione bent f del seguente tipo:

$$\begin{aligned} \lambda_0 =& \hat{f}(b(0)) = 2^{n-1} \pm 2^{\frac{n}{2}-1} = |\Omega_f|, \\ |\lambda_i| =& |\hat{f}(b(i))| = 2^{\frac{n}{2}-1}, \ i \neq 0. \end{aligned}$$

Concludiamo con un piccolo controesempio: sia $n = 2$ ed $f \in B_2$ la funzione $f(x_1, x_2) := x_1 x_2$. Si mostra che f è una funzione bent, infatti studiando lo spettro di W_f si ottiene $w = (2, -2, 2, 2)$ si ha $|w_i| = 2 = 2^{\frac{2}{2}}$. Il grafo associato G_f ha spettro degli autovalori invece dato da $Spec(G_f) = \{1, -1\}$ (infatti G_f è bipartito), che non ha però cardinalità uguale a 3, quindi non è fortemente regolare.

Riferimenti bibliografici

[1] Thomas W. Cusick e Pantelimon Stanica. *Cryptographic Boolean Functions and Applications*. 2009.

[2] *Fast Walsh–Hadamard transform*. URL: `https://en.wikipedia.org/wiki/Fast_Walsh%E2%80%93Hadamard_transform`.

[3] Audrey Terras. *Fourier Analysis on Finite Groups and Applications*. 1999.