Squares vs. rectangles: which ones are heavier?

Jérôme Plût

2013-12-03
Weighing squares and rectangles

Which one of these is heavier?

- Comparing areas is difficult!
- We compare the **Hamming weight** of their areas instead.
- We pick random squares and rectangles of size $2^N$.
- We compare squares and rectangles to lines (random numbers of size $2^{2N}$, with expected Hamming weight $N$).
Divide and conquer

On closer look, squares and rectangles have both a **big end** (top half) and a **small end** (bottom half):

![Illustration of big and small ends]

With boring numbers instead:

```
01101100001
```

**big end**  **small end**

This work is politically correct and inclusive. In particular, we respect all kinds of mathematics, and shall do analysis both in the **real numbers** (big end) and the **2-adic numbers** (small end).
On the small end

2-adic squares

A 2-adic number is a square iff it is of the form

$$(\ldots \ldots \ldots \text{00100} \ldots 00)_{\text{even}}.$$  

with geometric distribution.

- The expected weight of the lower half of a square is $-\frac{3}{2}$ bits.
- The expected weight of the lower half of a rectangle is $-\frac{1}{2}$ bits.

**The lower half of squares is lighter by 1 bit.**
Let $x \in [0, 2^N]$. 

- The $N$ bits on the big end of $x^2$ are the $N$ first bits of $(x/2^N)^2$.
- We know the density $f$ of $(x/2^N)^2$ in the interval $[0, 1]$:

$$f(t)dt = \frac{dt}{2\sqrt{t}}.$$ 

- We compute the average Hamming weight of a random number with density $f$. 
The Hamming weight \( S_n \) of \( t \in [0, 1] \) is the sum of the Hamming weight \( W_i \) of individual bits.

The functions \( W_i \) are periodic with period \( 2^{-i} \):

\[
W_1 : \quad \uparrow \quad \rightarrow \\
W_2 : \quad \uparrow \quad \rightarrow
\]

The expected value of \( W_i \) is

\[
\overline{W}_i = \int_0^1 W_i(t)f(t)dt = \langle W_i, f \rangle_{L^2}.
\]

We can compute this scalar product using Fourier series decomposition!
We compute the Fourier coefficients of $W_i$ and $f$ on $[0, 1]$.

Since we are lazy, we compute only $W_1$ and then $W_i(x) = W_1(2^{i-1}x)$.

Since we are lazy, we compute only the sine coefficients of $f$:

$$b_m(f) = 2 \int_0^1 \sin(2\pi mt) \frac{dt}{2\sqrt{t}} = \frac{2}{\sqrt{2\pi m}} \int_0^{\sqrt{2\pi m}} \sin(t^2) \, dt.$$
We compute the Fourier coefficients of $W_i$ and $f$ on $[0, 1]$.

Since we are lazy, we compute only $W_1$ and then $W_i(x) = W_1(2^{i-1}x)$.

Since we are lazy, we compute only the sine coefficients of $f$:

$$b_m(f) = 2 \int_0^1 \sin(2\pi mt) \frac{dt}{2\sqrt{t}} = \frac{\sqrt{2}}{\sqrt{2\pi} m} \int_0^{\sqrt{2\pi m}} \sin(t^2) \, dt.$$
How to get rid of @ll your annoying constants 1n one easy step

- We compute the Fourier coefficients of $W_i$ and $f$ on $[0, 1]$.
- Since we are lazy, we compute only $W_1$ and then $W_i(x) = W_1(2^{i-1}x)$.
- Since we are lazy, we compute only the sine coefficients of $f$:

$$b_m(f) = 2 \int_0^1 \sin(2\pi mt) \frac{dt}{2\sqrt{t}} = \frac{1}{\sqrt{2\pi m}} \int_0^{\sqrt{2\pi m}} \sin(t^2) dt.$$
We compute the Fourier coefficients of $W_i$ and $f$ on $[0, 1]$.

Since we are lazy, we compute only $W_1$ and then $W_i(x) = W_1(2^{i-1}x)$.

Since we are lazy, we compute only the sine coefficients of $f$:

$$b_m(f) = 2 \int_0^1 \sin(2\pi mt) \frac{dt}{2\sqrt{t}} = \frac{\sqrt{2\pi m}}{\sqrt{2\pi m}} \int_0^{\sqrt{2\pi m}} \sin(t^2) \, dt.$$
How to get rid of all your annoying constants in one easy step

- We compute the Fourier coefficients of $W_i$ and $f$ on $[0, 1]$.
- Since we are lazy, we compute only $W_1$ and then $W_i(x) = W_1(2^{i-1}x)$.
- Since we are lazy, we compute only the sine coefficients of $f$:

$$b_m(f) = 2 \int_0^1 \sin(2\pi mt) \frac{dt}{2\sqrt{t}} = \frac{\sqrt{2\pi m}}{\sqrt{2\pi m}} \int_0^{\sqrt{2\pi m}} \sin(t^2) \, dt.$$ 

- We get:

$$b_m(f) \sim \frac{1}{\sqrt{m}}.$$
Boringness is the cardinalest sing

- Sum the bits to compute the approximation for the Hamming weight of the first $n$ bits on the big end:

$$S_{n}^{sqr} = -1.5872394631649104531239363... + \frac{\sqrt{2} + 3}{2\pi} \zeta \left( \frac{3}{2} \right) 2^{-n/2} + \ldots$$

(gluttony) (pride) (sloth)
Boringness is the cardinalest sing

- Sum the bits to compute the approximation for the Hamming weight of the first $n$ bits on the big end:

\[
S^\text{sqr}_n = -1.5872394631649104531239363... + \frac{\sqrt{2} + 3}{2\pi} \zeta\left(\frac{3}{2}\right) 2^{-n/2} + \ldots
\]

(gluttony)  
(pride)  
(sloth)

- We may perform the same computations for rectangles...

\[
S^\text{mul}_n = -1.7289433... + \frac{\log 2}{2} \cdot n \cdot 2^{-n} + O(2^{-n}).
\]

(envy)

- The high half of squares is heavier by about 0.15 bit.
[Amiel, Feix, Tunstall, Whelan, Marnane 2008] observed that squares tended to be about 1 bit lighter than rectangles.

We wanted to determine the speed of convergence for increasing values of $n$.

We find that the average difference between the Hamming weight of a square and a product, as $n \rightarrow \infty$, is 0.8492962 bits.

So the actual speed of convergence to 1 bit is extremely slow.