Chapter 1: Significance of Entropy

December 23, 2003

Abstract

This chapter is about a method for generating solution to the Agarwal Entropy Hypothesis. It requires to be run on top of a method that calculates the probabilities at the output. Calculation of probabilities could be done, by sampling, or by using constructing Parker-Mclusky equations, or by using Seth-Agarwal PREDICT algorithm. Our method is equivalent to generic algorithms like gradient-descent, simulated-annealing that parse through the function space for a solution. The difference between generic algorithms and our method, is that we exploit properties of search-space, specific to binary functions.

1 Preliminaries

According to the Agarwal entropy hypothesis, the weightset $\mathbf{X}$ that optimally maximizes fault converage, is one, that is able to maximize output entropy function $\mathrm{H_o}(\mathbf{X})$, where $\mathrm{H_o}(\mathbf{X})$, is given by

$$\mathrm{E[C]}(\mathbf{X}) \log(\mathrm{E[C]}(\mathbf{X})) + (1 - \mathrm{E[C]}(\mathbf{X})) \log(1 - \mathrm{E[C]}(\mathbf{X})) \tag{1}$$

and $\mathrm{E[C]}(\mathbf{X})$ is the expected value of the circuit C with input weightset $\mathbf{X}$. In other words, a weightset which is able to set the probability of a output of the circuit to 0.5.

We extend the Agarwal entropy hypothesis, and add an extra condition. Our approach is based on Agarwal's orginal reasoning. A best weight $\mathbf{X}$ is one, that is able to both increase both input entropy and output entropy. Input entropy function $\mathrm{H_i}(\mathbf{X})$, is given by

$$\sum_{i=1}^{n} x_i \log(x_i) + (1 - x_i) \log(1 - x_i) \tag{2}$$

To understand our conclusion, suppose $\mathrm{H_o}$ is the output entropy and $\mathrm{H_i}$ is the input entropy,

$$\frac{\mathrm{H_o}}{\mathrm{H_i}} \tag{3}$$

is the probability that information, at the input, reaches the output. Virtue of input entropy being the average amount of information generated at input and the output entropy being the average amount of information received at the output, $\frac{\mathrm{H_o}}{\mathrm{H_i}}$ becomes the ratio(probability) that a bit generated at the input, is received at the output.
The better this metric[1] ratio, more paths are able to propogate a fault to the output. Concepts of entropy are related to reversibility. If you can put in H_i and get an output of $\mathrm{H_i}$,now based on the output, you can completely get back the input. Now, if you put in $\mathrm{H_i}$ and get an output of $\mathrm{H_i}/2$, based on the information at the output, you cannot exactly guess back the input. The better the ability to reverse,the better,one is able to sensitize a fault.
Now,

$$\frac{\mathrm{H_i}}{\mathrm{n}} \tag{4}$$

is the proportional to the amount of information, generatable at the input. The better this metric, more number of faults are assessible to changes in the input.

2 Initial Value

Suppose, one tries to set a circuit's output to 0.5. And suppose, when the inputs of the circuits are all set to 0.5, the probability at the output, of the circuit, is also less than or equal to 0.5, we can do the following

Now, suppose we find a binary input pattern that is able to set the circuit to 1. Let n, be the number of inputs in the circuit.

We take $c = 0.5^{1/n}$. We replace the zeros of the input pattern with $1 - c$ and replace the ones of the input pattern with c. Now, the pattern, by itself, supplies, a probability value of 0.5. Now, Remember that, the probability at any of the inputs, will be either c or 1-c.
Now, to understand why the method works, we will try to walk through the bionimal expansion of $(a + b)^n$.

$$(a+b)^n = C_0^n a^n + C_1^n a^{n-1} b^1 + C_2^n a^{n-2} b^2 + ... + b^n \tag{5}$$

Now, one of the standard ways, to intrepreted that , is that nC_k is the number of times the term $a^k b^{n-k}$ shows up, in the expansion. Similarly, suppose we expand $(c + 1 - c)^n$. Now $(c + 1 - c)^n = 1$, is trivally equal to 1. But, notice the following, about its expansion.

$$\begin{aligned}(c+1-c)^n &= c^n + C_1^n c^{n-1}(1-c) + C_2^n c^{n-2}(1-c)^2 + ... && (6)\\ & C_k^n c^{n-k}(1-c)^k + ... && (7)\\ &= c^n + nc^{n-1}(1-c) + \frac{n(n-1)}{2!}c^{n-2}(1-c)^2 + ... && (8)\end{aligned}$$

[1]We use metric,in the sense,that this quantity provides insight into the actual quantity,by their relationships could be complex.

$$+\frac{n(n-1)(n-2)...(n-k+1)}{k!}c^{n-k}(1-c)^k + ... \quad (9)$$

$$(10)$$

Now, $c^n = 0.5$ because $c = 0.5^{1/n}$. Now as n gets large, c^{n-2}, c^{n-3},etc, will stay close to 0.5.
Take expression $nC_k(1-c)^k$. Take c = $0.5^{1/n}$. Now, $c = e^{\frac{\ln 0.5}{n}}$. Now, as n gets large, we can approximate c, very closely, by only taking the first-order approximation of $\exp(x)$.

$$\exp(x) = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + ... \quad (11)$$

Now as $n \to \infty$, the expansion for $c = e^{\frac{\ln 0.5}{n}}$ can be writtern as

$$c \approx 1 + \frac{\ln 0.5}{n} \quad (12)$$

Now

$$1 - c = -\frac{\ln 0.5}{n} = \frac{\ln 2}{n} \quad (13)$$

If n is large enough, we have

$$nC_k(1-c)^k = \frac{n^k}{k!}\frac{\ln^k 2}{n^k} = \frac{\ln^k 2}{k!} \quad (14)$$

So, $(c + 1 - c)^n$ is

$$= \quad 0.5 + \tfrac{\ln 2}{1!}(0.5) + \tfrac{\ln^2 2}{2!}(0.5) + \tfrac{\ln^3 2}{3!}0.5 + ... \quad (15)$$

What is important about this result is that it says, just about n^3 terms, with our condition on c, contribute to 99.4% of the probabilities. We do not have to bother, with the rest of the terms because, they can atmost contribute to .6% of the probability.

3 Effective Approximation

Now, first, it must be noticed that, by picking a vector that contributes to 0.5 of the probability, the vectors that contribute to rest of the higher order terms, are fixed.
Let $\frac{K}{2^n}$ be the average probability,that a variation of the most-contributing vector, is part of the function.

Now, take the expansion

$$= \quad 0.5 + \tfrac{\ln 2}{1!}(0.5) + \tfrac{\ln^2 2}{2!}(0.5) + \tfrac{\ln^3 2}{3!}0.5 + ... \quad (16)$$

We are certain about the 0.5. The rest of the higher order terms, come from vectors, each with a probability of $\frac{K}{2^n}$, of being part of the function.
For the given c, if $\frac{K}{2^n}$ converges to a single number, the expansion can be safely written as,

$$= \quad 0.5 + \frac{K}{2^n}\left(\frac{\ln 2}{1!}(0.5) + \frac{\ln^2 2}{2!}(0.5) + \frac{\ln^3 2}{3!}0.5 + ...\right) \tag{17}$$

The smaller $\frac{K}{2^n}$, the better the approximation. Generally, $\frac{K}{2^n}$ is equal to the $\mathrm{E[C]}(x_i = 0.5)$

This approximation can be made better, by estimating the effective probability $\frac{K_{\text{eff}}}{2^n}$. To do that, take

$$\frac{K_{\text{eff}}}{2^n} = \frac{\mathrm{E[C]}[x_i = 0.5^{1/n}] - 0.5}{0.5} \tag{18}$$

Proof 1 *Start by assuming*

$$\mathrm{E[C]}(x_i = 0.5^{1/n}) = 0.5 + \frac{K_{\text{eff}}}{2^n}\left(\frac{\ln 2}{1!}(0.5) + \frac{\ln^2 2}{2!}(0.5) + \frac{\ln^3 2}{3!}0.5 + ...\right) \tag{19}$$

Now,

$$1 = 0.5 + \frac{\ln 2}{1!}(0.5) + \frac{\ln^2 2}{2!}(0.5) + \frac{\ln^3 2}{3!}0.5 + ... \tag{20}$$

Substituting that, we would get

$$\mathrm{E[C]}(x_i = 0.5^{1/n}) = 0.5 + \frac{K_{\text{eff}}}{2^n}0.5 \tag{21}$$

In general, for a given value, d, the expansion can be written as

$$= \quad d + \frac{K_{\text{eff}}}{2^n}\left(-\frac{\ln d}{1!} + \frac{\ln^2 d}{2!} - \frac{\ln^3 d}{3!} + ...\right)d \tag{22}$$

4 Finding a Solution

Instead of trying to solve a complicated n-dimensonal problem, we can convert it into a 1-dimensional problem, in the following sense.

Suppose, you find a binary input vector that is able to set the output of the function to a 1, and suppose the complement of the binary input vector, is able to set the function to a 0. Now, here's something interesting and important.
Suppose, if the natural output probability of the circuit is less than 0.5, we replace the 1s of the vector with c and 0s of the input vector with $1 - c$.When $c = 1$, it will set the output of the function to a 1, and when $c = 0$, the output of the function becomes a 0. Then according to the intermediate value theorem, there is a c in the interval [0,1], that will give you a 0.5. Now, we can make our

initial guess, from the methods described in the above section and improve the solution, using a nice root finding method.

If you can satisfy the conditions on c, you will always find a solution,using a good root-finding procedure. To calculate your luck of guessing a solution correctly, consider the following arguement. Suppose, the 1s and the 0s of the function are statiscally unrelated. Now, if the probability of being a 1 is ϕ. The probabiliy that the function is 0 is $1-\phi$. If they are statiscally unrelated, then, $\phi(1-\phi)$ is **approximately** the probability of finding an input vector, that is able to set the function to 1 and whose complement is able to set the function to 0. Now, as ϕ goes to 1 or 0, the probability of guessing a good input-vector candidate, decreases.

4.1 An Ideal Iterator

A very successful iterator candidate, that can quickly converge onto the correct c, is

$$\frac{K_{\text{eff}}}{2^n} = \frac{\text{E[C]}(x_i = c_{t-1}) - c_{t-1}^n}{1 - c_{t-1}^n} \tag{23}$$

And

$$c_t = \sqrt[n]{\left| \frac{0.5 - \frac{K_{\text{eff}}}{2^n}}{1 - \frac{K_{\text{eff}}}{2^n}} \right|} \tag{24}$$

Proof 2 *Take the expansion*

$$d + \frac{K_{\text{eff}}}{2^n} \left(-\frac{\ln d}{1!} + \frac{\ln^2 d}{2!} - \frac{\ln^3 d}{3!} + ... \right) d \tag{25}$$

Idealy, this must equal $\text{E[C]}(x_i = c)$. *Assuming that, we can estimate* $\frac{K_{\text{eff}}}{2^n}$, *substituting*

$$1 - d = \left(-\frac{\ln d}{1!} + \frac{\ln^2 d}{2!} - \frac{\ln^3 d}{3!} + ... \right) d \tag{26}$$

Now, assuming that $d = c_{t-1}^n$

$$\text{E[C]}(x_i = c_{t-1}) = c_{t-1}^n + \frac{K_{\text{eff}}}{2^n} \left(1 - c_{t-1}^n\right) \tag{27}$$

Rewriting this step, would give the expression for $\frac{K_{\text{eff}}}{2^n}$ *as*

$$\frac{K_{\text{eff}}}{2^n} = \frac{\text{E[C]}(x_i = c_{t-1}) - c_{t-1}^n}{1 - c_{t-1}^n} \tag{28}$$

Suppose, we our estimate is correct

$$d + \frac{K_{\text{eff}}}{2^n}\left(-\frac{\ln d}{1!} + \frac{\ln^2 d}{2!} - \frac{\ln^3 d}{3!} + ...\right) d \tag{29}$$

must equal 0.5.
From this step, we can estimate c_t *, by substituting* $d = c_t^n$

$$c_t^n + \frac{K_{\text{eff}}}{2^n}(1 - c_t^n) = 0.5 \tag{30}$$

Rewriting that, would give us

$$c_t = \sqrt[n]{\left|\frac{0.5 - \frac{K_{\text{eff}}}{2^n}}{1 - \frac{K_{\text{eff}}}{2^n}}\right|} \tag{31}$$

4.2 Conditions and Convergence

It must be noted that, as we try to converge output onto 0.5, input probability $d^{1/n}$ will shift more and more towards 0.5, and more and more terms will be able to play a major role in the calculation of the output probability

$$= \quad d + \tfrac{K_{\text{eff}}}{2^n}\left(-\tfrac{\ln d}{1!} + \tfrac{\ln^2 d}{2!} - \tfrac{\ln^3 d}{3!} + ...\right) d \tag{32}$$

This is good, since, the closer $d^{1/n}$ towards 0.5, the more vectors, we have to choose from. The pool of available amount of vectors can be measured by input entropy.

$$\sum_{i=1}^{n} x_i \log(x_i) + (1 - x_i)\log(1 - x_i) \tag{33}$$

Convergence of $\frac{K_{\text{eff}}}{2^n}$ is equivalent to the convergence of c.

5 The Diffractor

Our solution is a diffraction pattern,nontheless. Now, suppose our circuit function can be represented by C($\mathbf{X}$). Now, in the above section, we gave details on converting a pattern to a solution, by replacing 1s of the pattern with c, and 0 of the pattern with $1 - c$, when probability of the function being a 1, is less than 0.5.
Suppose we replace C($\mathbf{X}$) with C($\mathbf{Y}$), where y_i is $1 - x_i$, when the bit i, of the pattern is 0, or with x_i, when the bit i, of the pattern is 1.
Let's take the input entropy function $\mathrm{H_i}(\mathbf{X})$ be defined as

$$\sum_{i=1}^{n} x_i \log(x_i) + (1 - x_i)\log(1 - x_i) \tag{34}$$

The function would not change if we replace it, with y_i. So, we define, entropy function $\mathrm{H_i}(\mathbf{Y})$ be defined as

$$\sum_{i=1}^{n} y_i \log(y_i) + (1 - y_i) \log(1 - y_i) \tag{35}$$

Now, the output entropy function $\mathrm{H_o}(\mathbf{Y})$, for modified $\mathbf{X}$, be defined as

$$\mathrm{E[C]}(\mathbf{Y}) \log(\mathrm{E[C]}(\mathbf{Y})) + (1 - \mathrm{E[C]}(\mathbf{Y})) \log(1 - \mathrm{E[C]}(\mathbf{Y})) \tag{36}$$

It should be noted that when the output probability is 0.5, the entropy there is already maximum.
Now we shall prove that

$$\frac{\partial \mathrm{H_i}}{\partial y_i} = \lambda \frac{\partial \mathrm{H_o}}{\partial y_i} \tag{37}$$

This step is the lagrange multiplier way, of showing that input and output entropy are locally simulatenously maximized.
Differentiating H_i with respect to y_i, we get

$$\frac{\partial \mathrm{H_i}}{\partial y_i} = \log\left(\frac{y_i}{1 - y_i}\right) \tag{38}$$

Differentiating H_o with respect to y_i, we get

$$\frac{\partial \mathrm{H_o}}{\partial y_i} = \log\left(\frac{\mathrm{E[C]}(\mathbf{Y})}{1 - \mathrm{E[C]}(\mathbf{Y})}\right) \frac{\partial \mathrm{E[C]}}{\partial y_i}(\mathbf{Y}) \tag{39}$$

By virtue of

$$\log\left(\frac{\mathrm{E[C]}(\mathbf{Y})}{1 - \mathrm{E[C]}(\mathbf{Y})}\right) = 0 \tag{40}$$

since $\mathrm{E[C]}(\mathbf{Y}) = 0.5$ any such input probability vector , will locally maximize both input and output entropy, when the output is set to 0.5.
In the above section, we utilized the fact that $\log(0.5/0.5) = 0$. It should specifically be noted that, the result would still hold, even that is not the case. Take the expression

$$d + \frac{k_{eff}}{2^n}(1 - d) \tag{41}$$

Now, suppose, we differentiate d with respect to x_i, we will get

$$d^{\frac{n-1}{n}} + \frac{k_{eff}}{2^n}(1 - d^{\frac{n-1}{n}}) \tag{42}$$