From Linear Recurrences to Random Signals, and Back

Sailesh Simhadri[1] (advised by Iddo Ben-Ari[2])

[1] Department of Computer Science and Engineering, University of Connecticut, Storrs, CT, USA

[2] Department of Mathematics, University of Connecticut, Storrs, CT, USA

Linear Recurrence Enumeration Systems

Decomposition of positive integers as linear combinations of elements in a linear recurrence.

Consider a linear recurrence of length L with nonnegative integer coefficients:
$G_{n+1} = c_1 G_n + \dots c_n G_{n-L+1}$ with $G_1=1, G_2=2, \dots, G_{L-1}=L-1, G_L=L$

Examples.

- Decimal: $G_{n+1} = 10G_n$.
  - $11 = 1{*}G_2 + 1{*}G_1$
- Binary: $G_{n+1} = 2\,G_n$.
  - $11 = 1{*}G_4 + 0{*}G_3 + 1{*}G_2 + 1{*}G_1 \Rightarrow 1011$
- Zeckendorf: $G_{n+1} = G_n + G_{n-1}$ (Fibonacci sequence, L=2)
  - $G_1=1, G_2=2, G_3=3, G_4=5, G_5=8, \dots$
  - $11 = 1{*}G_5 + 0{*}G_4 + 1{*}G_3 + 0{*}G_2 + 0{*}G_1 \Rightarrow 10100$

Theorem. Every positive integer n, there exists a unique decomposition $m = X_1 G_n + X_2 G_{n-1} + \dots X_n G_1$, "dominated" by the recurrence.

Proof. Greedy algorithm, refer to [Miller and Wang 2012] and the references within.

Question

Looking at a long sequence from a random source, can you tell if the source is a Zeckendorf? If so, what is the recurrence?

Theorem

Definition.
Let R_1 denote the recurrence relation
$G_{n+1} = c_1 G_n + \dots + c_L G_{n-L+q}$, and let R_2 denote the recurrence relation
$G_{n+1} = d_1 G_n + \dots + d_K G_{n-K+1}$, where, without loss of generality, $K \geq L$.
We say that R_2 and R_1 are equivalent if

1) If K is an integer multiple of L; and
2) $(d_1,\dots,d_K) = (c_1,\dots,c_L-1,\dots,c_1,\dots,c_L-1,c_1,\dots,c_L)$

The equivalence class of R_1 is all recurrence relations equivalent to R_1.

Example.
R_1: $G_{n+1} = 1{*}G_n + 1{*}G_{n-1}$ and R_2: $G_{n+1} = 1{*}G_n + 0{*}G_{n-1} + 1{*}G_{n-2} + 1{*}G_{n-3}$ are equivalent.

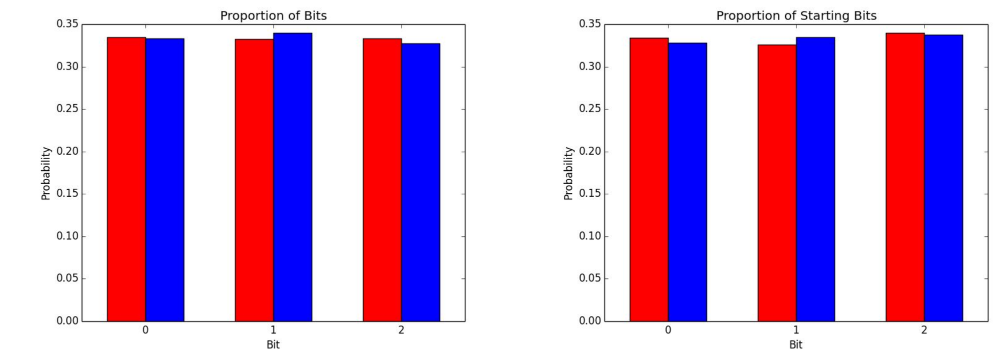
Theorem [Ben-Ari, Simhadri]

a) If R_1 and R_2 are equivalent then signals from either sources are indistinguishable.
b) Given a source corresponding to some recurrence relation, there exists an algorithm that uniquely determines its equivalence class.

Motivating Example

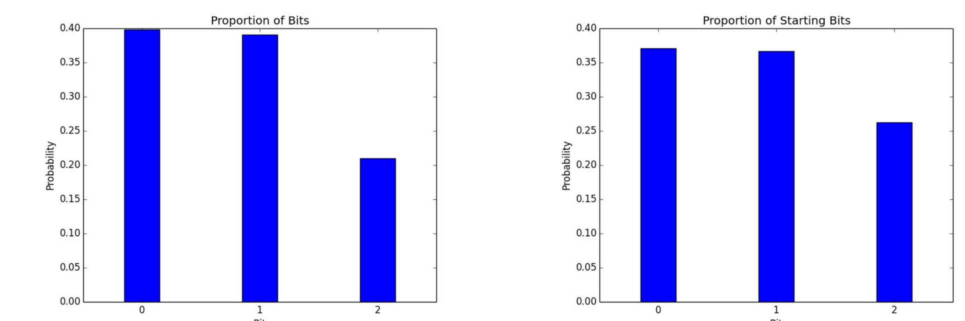
$G_n=2G_{n-1}+3G_{n-2}$

$G_n=3G_{n-1}$



The graphs above are from simulations of equivalent recurrences. Note how the proportions of bits and starting bits coincide. This is obvious for the second recurrence, and holds for the first one due to the equivalence.

$G_n=2G_{n-1}+2G_{n-2}$



Although sequences generated by this recurrence are all legal for the pair of equivalent recurrences above, the statistics are clearly different. Also, the proportions of bits and starting bits no longer coincide.

Random Source

Random source.
Sample uniformly a number m in $[G_n, G_{n+1})$ for some large n, yielding a random process $X=(X_1,X_2,\dots,X_n)$ of the coefficients (c_i) in the decomposition of m.

Example.
Zeckendorf decomposition.
Sampling uniformly between $[G_2, G_3) = [3, 5)$. 3=0100, 4=0101
Then, random sequence is either 0100 or 01010. We will be looking at sequences for a larger n.

In general, the resulting process is not IID, and not even Markovian.

Example.
Consider the recurrence relation $G_{n+1} = 2G_n + 2G_{n+1} + G_{n-2}$
If $X_j=2$ then, depending on where j is, X_{j+1} may take any value in {0,1,2} or only the value 0.

Legal Sequences.

- Start from j=0 (beginning of X) and i=1 (position within a word).
- No X_j is bigger than any c_i.
- If $X_j < c_i$, set i to one, starting a new word.
- There is never an instance of consecutive c_1 to c_n in the sequence.

Algorithm

We assume our source is some unknown recurrence.

For i=0,1,2,... (while sequence is legal with respect to current guessed recurrence)

a. Determine c_i by looking at the maximal element after multiple zeros followed by $c_1, c_2, \dots, c_{i-1}$ ($000c_1c_2 \dots c_{i-1}$). Check consistency by seeing if the recurrence $c_1,c_2,\dots,$**(c_i+1)** can produce the output legally. If not, repeat loop to find c_{i+1}.
b. Generate random sequences of the source corresponding to the recurrence. $c_1,c_2,\dots,$**(c_i+1)**. See if the proportions of c_j for j=1,...,i are the same for this generated sequence as the experimental. If not, repeat loop to find c_{i+1}.
c. Return the recurrence $c_1,c_2,\dots,$**(c_i+1)**.

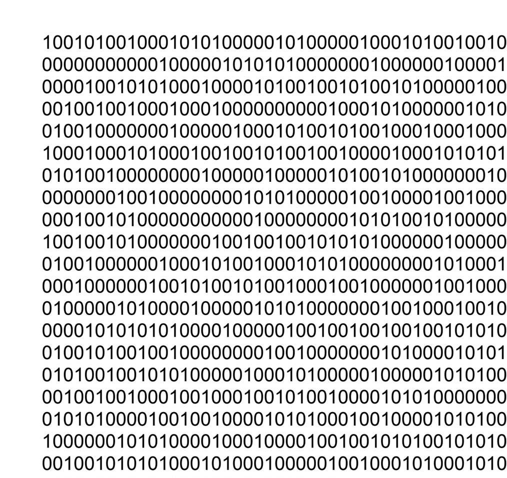
Note: There is a faster algorithm for monotonic sequences that does not rely on the usage of probabilities.

Example.
X = 1**00**1010010**00**1010100

1. After multiple zeros (bolded), there is a 1 $\Longrightarrow c_1=1$
2. Consistent with $c_1=2$.
3. Probability check fails.
4. After multiple zeros and 1, the next element is 0. Therefore, $c_2=0$. (Add 1 in next step)
5. Sequence is consistent with $c_1=1$, $c_2=1$
6. Probability check passes, so return recurrence.

Why Zeckendorf

- Sequences have more structure, allowing for better error detection.
- More relevant for modeling signals with inherent structure. Models constraints or "grammar":
  - Never have two consecutive ones (Zeckendorf), see below.



A sequence generated from a source given by the Zeckendorf decomposition. The following was first proved in [Lekkerkerker]:

The (asymptotic) proportion of ones is $1/(\phi+2) \sim= .2764$

As a result, the proportion of words starting with one is $1/(\phi+1) \sim= .382$

Where $\phi = (1+\sqrt{5})/2$ is the Golden Ratio.

UCONN
UNIVERSITY OF CONNECTICUT