Taking the "Convoluted" out of Bernoulli Convolutions

A Combinatorial Approach

Michelle Delcourt

Additive Combinatorics Mini-Conference
Georgia Tech

June 26, 2010

Neil Calkin, Julia Davis, **Michelle Delcourt,** Zebediah Engberg, Jobby Jacob, and Kevin James. Discrete Bernoulli Convolutions: An Algorithmic Approach Toward Bound Improvement. *To appear, Proceedings of the AMS.*

Motivation

Bernoulli Convoluted

A Bernoulli Convolution is the convolution

$$\mu_q(X) = b(X) * b(X/q) * b(X/q^2) * \ldots$$

where b is the discrete Bernoulli measure concentrated at 1 and -1 each with weight $\frac{1}{2}$.

Consider the distribution function we define as $F_q(t) = \mu_q([-\infty, t])$.

Functional Equation

Instead consider the functional equation

$$F(t) = \frac{1}{2}F\left(\frac{t-1}{q}\right) + \frac{1}{2}F\left(\frac{t+1}{q}\right)$$

for t on the interval $I_q := \left[\frac{-1}{(1-q)}, \frac{1}{(1-q)}\right]$.

It can be shown that there is a unique continuous solution $F_q(t)$ to the above equation.

Absolutely Continuous vs. Singular

The major question regarding the solution of the previous equation is that of determining the values of q that make $F_q(t)$ absolutely continuous and the values that make $F_q(t)$ singular.

When $0 < q < \frac{1}{2}$, Kershner and Wintner have shown that $F_q(t)$ is always singular. For these values of q, the solution $F_q(t)$ is an example of a so-called *Cantor function*, a function that is constant almost everywhere.

It is also easy to see that for $q = \frac{1}{2}$, the solution $F_q(t)$ is absolutely continuous.

Pisot Numbers

The case when $q > \frac{1}{2}$ is much harder and more interesting.

In 1939, Erdős showed that if q is of the form $q = \frac{1}{\theta}$ with θ a *Pisot number*, then $F_q(t)$ is again singular.

A *Pisot number* is an algebraic integer greater than 1 in absolute value, whose conjugates are all less than 1 in absolute value.

The classic example is the golden ratio $\varphi = \frac{(1+\sqrt{5})}{2}$. Like all Pisot numbers, φ has the property that large powers of φ approach rational integers.

No Actual Example is Known

There is little else that is known for other values of $q > \frac{1}{2}$.

One interesting result due to Solomyak is that almost every $q > \frac{1}{2}$ yields a solution $F_q(t)$ that is absolutely continuous.

Hence it is surprising that no actual example of such a q is known.

Specifically, the obvious case when $q = \frac{2}{3}$ remains a mystery.

Differentiate

Rather than looking at the function $F_q(t)$, one can also consider its derivative $f_q(t) = F_q'(t)$. Upon differentiating, the functional equation for $F_q(t)$ gives the following equation for $f_q(t)$:

$$f(t) = \frac{1}{2q} f\left(\frac{t-1}{q}\right) + \frac{1}{2q} f\left(\frac{t+1}{q}\right).$$

The question of the existence of an absolutely continuous solution $F_q(t)$ to the previous equation is equivalent to the existence of an $L^1(I_q)$ solution $f_q(t)$ to the above equation.

A New Functional Equation

Girgensohn asks the question of computing $f_q(t)$ for various values of q. The author considers starting with an arbitrary initial function $f_0(t) \in L^1(I_q)$ and iterating the transform

$$T_q : f(t) \longmapsto \frac{1}{2q} f\left(\frac{t-1}{q}\right) + \frac{1}{2q} f\left(\frac{t+1}{q}\right)$$

to gain a sequence of functions $f_0, f_1, f_2, \ldots$. If this sequence converges, then it converges to the solution of the previous functional equation.

$$F(t) = \frac{1}{2} F\left(\frac{t-1}{q}\right) + \frac{1}{2} F\left(\frac{t+1}{q}\right)$$

$q = 2/3$

Neil Calkin then looked at the above process for $q = \frac{2}{3}$. Rather then working on the interval $I_q = [-3, 3]$, we shift the entire interval to $[0, 1]$ for simplicity.

The transform T_q now becomes the transform $T : L^1([0, 1]) \longrightarrow L^1([0, 1])$ where

$$T : f(x) \longmapsto \frac{3}{4} f\left(\frac{3x}{2}\right) + \frac{3}{4} f\left(\frac{3x-1}{2}\right).$$

The Transform

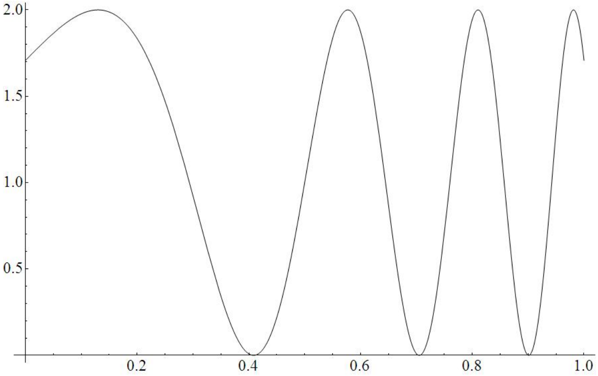
Intuitively, this transform takes two scaled copies of $f(x)$: one on the interval $\left[0, \frac{2}{3}\right]$ and the other on $\left[\frac{1}{3}, 1\right]$, and adds them.

The scaling factor of $\frac{3}{4}$ gives us that

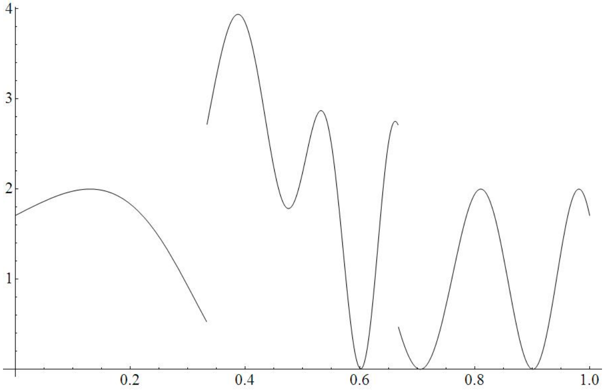
$$\int_0^1 f(x)dx = \int_0^1 Tf(x)dx.$$

In this setting, the question to be answered is: starting with the function $f_0(x) = 1$, does the iteration determined by this transform converge to a bounded function?

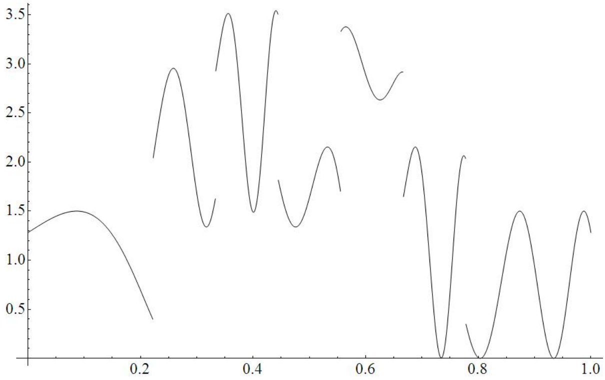
f_0



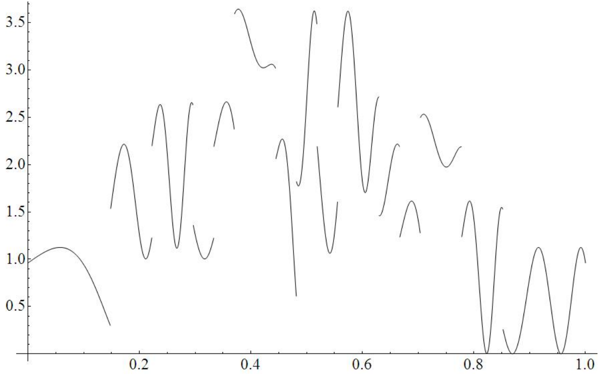
f_1



f_2



f_3



A Recursive Algorithm

Duplicate, Shift, Add

Duplicate, Shift, Add

Instead of viewing T as a transform on $[0, 1]$, we consider the combinatorial analogue described by the sequences $\mathrm{dup}(B_n)$ and $\mathrm{shf}(B_n)$. Consider the two maps $\mathrm{dup}_n, \mathrm{shf}_n : \mathbb{R}^n \longrightarrow \mathbb{R}^{3n}$ defined by

$$\mathrm{dup}_n : (a_1, a_2, \ldots, a_{n-1}, a_n) \longmapsto (a_1, a_1, a_2, a_2, \ldots, a_{n-1}, a_{n-1}, a_n, a_n, \overbrace{0, \ldots, 0}^{n \text{ times}})$$

$$\mathrm{shf}_n : (a_1, a_2, \ldots, a_{n-1}, a_n) \longmapsto (\overbrace{0, \ldots, 0}^{n \text{ times}}, a_1, a_1, a_2, a_2, \ldots, a_{n-1}, a_{n-1}, a_n, a_n).$$

Duplicate, Shift, Add

The names “dup” and “shf” reference the duplication and shifting of the coordinates.

Consider the finite sequences of increasing length given by $B_0 = (1)$ and $B_{n+1} = \mathrm{dup}_n(B_n) + \mathrm{shf}_n(B_n)$.

We are primarily interested in the rate at which m_n, the maximum of B_n, is growing with n.

Duplicate, Shift, Add

The fact that B_n has a total of 3^n terms follows directly from the definition of dup_n and shf_n.

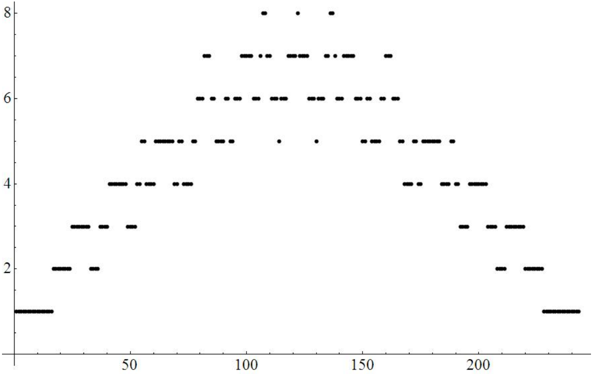
$$\begin{array}{r} 1 \rightarrow 1\ 1 \\ 1\ 1 \\ \hline 1\ 2\ 1 \end{array}$$

$$\begin{array}{r} 1\ 2\ 1 \rightarrow 1\ 1\ 2\ 2\ 1\ 1 \\ 1\ 1\ 2\ 2\ 1\ 1 \\ \hline 1\ 1\ 2\ 3\ 2\ 3\ 2\ 1\ 1 \end{array}$$

$$\begin{array}{r} 1\ 1\ 2\ 3\ 2\ 3\ 2\ 1\ 1 \rightarrow 1\ 1\ 1\ 1\ 2\ 2\ 3\ 3\ 2\ 2\ 3\ 3\ 2\ 2\ 1\ 1\ 1\ 1 \\ 1\ 1\ 1\ 1\ 2\ 2\ 3\ 3\ 2\ 2\ 3\ 3\ 2\ 2\ 1\ 1\ 1\ 1 \\ \hline 1\ 1\ 1\ 1\ 2\ 2\ 3\ 3\ 2\ 3\ 4\ 4\ 3\ 4\ 3\ 4\ 4\ 3\ 2\ 3\ 3\ 2\ 2\ 1\ 1\ 1\ 1 \end{array}$$

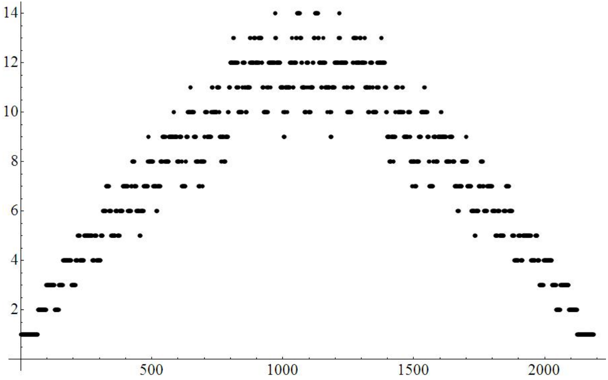
Level $n = 5$

The plot shows the index on the horizontal axis and the B_5 entry on the vertical axis.



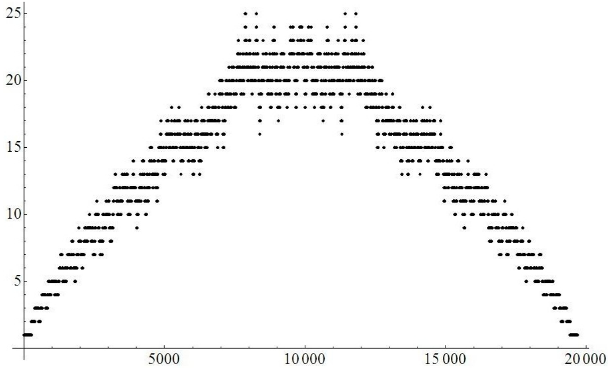
Level $n = 7$

The plot shows the index on the horizontal axis and the B_7 entry on the vertical axis.



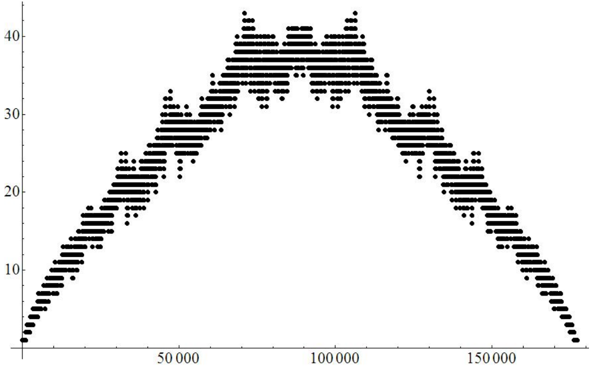
Level $n = 9$

The plot shows the index on the horizontal axis and the B_9 entry on the vertical axis.



Level $n = 11$

The plot shows the index on the horizontal axis and the B_{11} entry on the vertical axis.



A Useful Property

It is straightforward to see that the mean $\mu(B_n) = \left(\frac{4}{3}\right)^n$.

The reason for this is because under the DSA process, the length of a Bernoulli sequence grows by a factor of three while the sum of the terms increases by a factor of four.

Does m_n also grow like $\left(\frac{4}{3}\right)^n$?

Translating DSA as a Polynomial Recursion

Polynomial Approach

Translating DSA as a Polynomial Recursion

By encoding these sequences as coefficients of polynomials, the process of *duplicate, shift, add* gives a particularly nice recursive relation among the polynomials.

Let $B_n = (b_0, b_1, ..., b_t)$ be the Bernoulli sequence on level n where $t = 3^n - 1$.

Consider the polynomial $p_n(x) := b_0 + b_1 x + ... + b_t x^t$.

Translating DSA as a Polynomial Recursion

We see that the duplication $b_0, b_0, b_1, b_1, ..., b_t, b_t$ corresponds to the polynomial $(1+x)p_n(x^2)$.

Shifting the sequence 3^n places to the right corresponds to multiplication by x^{3^n}.

By adding the duplicate and the shift of the sequence, we yield the recurrence relation:

$$p_{n+1}(x) = (1+x)p_n(x^2)\left(1+x^{3^n}\right).$$

Translating DSA as a Polynomial Recursion

This formula allows us to explicitly solve for $p_n(x)$.

Theorem

(Calkin) The polynomials $p_n(x)$ satisfy

$$p_n(x) = \prod_{i=0}^{n-1} \left(1 + x^{2^i}\right) \prod_{j=0}^{n-1} \left(1 + x^{2^{n-1}(3/2)^j}\right).$$

The proof follows by induction on n.

A Bound on the Coefficients

Theorem

(Calkin) The maximum values satisfy $m_n = O((\sqrt{2})^n)$.

By factoring p_n in a clever way, we can put a bound on how fast the coefficients grow with the level n.

PIP

Polynomial Isolated Point

A Non-recursive Approach

An Algorithmic Implementation: PIP

Our algorithm is based on the following idea. Suppose $S = \{a_1, ..., a_n\}$ is a sequence of positive integers. Consider the polynomial

$$f(x) = \prod_{i=1}^{n} (1 + x^{a_i}) = \sum_{j=1}^{m} \alpha_j x^j.$$

Then α_j is the number of ways to write j as a sum of distinct elements from S.

This idea is applicable to the coefficients of our polynomial because our polynomial is a product of terms of the form $(1 + x^a)$.

An Algorithmic Implementation: PIP

Hence we get that the coefficient b_j of x^j in $p_n(x)$ (which is the j^{th} entry on the n^{th} Bernoulli sequence) is precisely the number of ways that j can be written as a sum of distinct terms in the sequence

$$S = \{1, 2, 4, ..., 2^{n-2}, 2^{n-1}, 2^{n-1}, 2^{n-2}3, 2^{n-3}3^2, ..., 2^2 3^{n-3}, 2^1 3^{n-2}, 3^{n-1}\}.$$

An Algorithmic Implementation: PIP

We now outline an algorithm that can be used to calculate the entry b_j for a fixed level n. The entire algorithm is based on the following ideas:

- Let $S = \{a_1, ..., a_n\}$ where each $a_i > 0$. Let $\mathbb{N}_S(k)$ denote the number of ways to write k as a sum of distinct elements from S. Then for any $i \in \{1, ..., n\}$, the following holds

$$\mathbb{N}_S(k) = \mathbb{N}_{S\setminus\{a_i\}}(k) + \mathbb{N}_{S\setminus\{a_i\}}(k - a_i).$$

- If $k > \sum_{s \in S} s$, then $\mathbb{N}_S(k) = 0$.

An Algorithmic Implementation: PIP

- If $k < 0$, then $\mathbb{N}_S(k) = 0$.

- We see that if $0 < k < 2^{n-1}$, then $\mathbb{N}_S(k) = \mathbb{N}_{S'}(k)$ where $S' = \{1, 2, 4, ..., 2^{n-2}\}$ since all other elements of S are too large. However, every k with $0 < k < 2^{n-1}$ can be written uniquely as a sum from elements of S'; this is simply the binary expansion of k. Our "cutoff" value is $2^{n-1} - 1$.

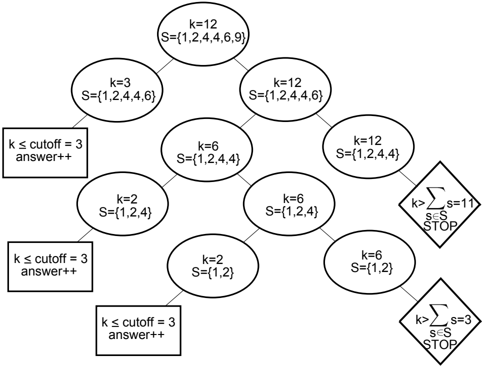
Flowchart

The following flowchart shows an explicit example of our implementation of the PIP algorithm in use.

It shows the computation of $\mathbb{N}_S(k)$ for $k = 12$ and $n = 3$. $S = \{1, 2, 4, 4, 6, 9\}$. As the diagram indicates, $\mathbb{N}_S(k) = 3$, corresponding to the fact that there are three boxes containing the word *answer* $++$

$B_3 =$
$1, 1, 1, 1, 2, 2, 3, 3, 2, 3, 4, 4, \color{red}{3}, 4, 3, 4, 4, 3, 2, 3, 3, 2, 2, 1, 1, 1, 1$

Flowchart


k=12
S={1,2,4,4,6,9}
k=3
S={1,2,4,4,6}
k=12
S={1,2,4,4,6}
k ≤ cutoff = 3
answer++
k=6
S={1,2,4,4}
k=12
S={1,2,4,4}
k=2
S={1,2,4}
k=6
S={1,2,4}
k>∑ s=11
s∈S
STOP
k ≤ cutoff = 3
answer++
k=2
S={1,2}
k=6
S={1,2}
k ≤ cutoff = 3
answer++
k>∑ s=3
s∈S
STOP

Better Bound

Normalize

Seeing that our sequence on level n has length 3^n, we naturally index it by the first 3^n nonnegative integers.

In certain circumstances, it is advantageous to normalize the indexing in such a way that each index is on the interval $[0, 1]$.

To this end, we can simply take the image of $k \in \{0, 1, 2, ..., 3^n - 1\}$ under the map $k \mapsto k/3^n$.

Notation

To emphasize our new indexing scheme, let $g_n(x)$ denote the n^{th} level Bernoulli sequence where now $x \in [0,1]$. In other words,

$$g_n\left(\frac{k}{3^n}\right) = b_k \qquad \text{for } k = 0, 1, \ldots 3^n - 1.$$

For a subset $S \subset [0,1]$, we define

$$m_n(S) = \max_{x \in S} g_n(x)$$

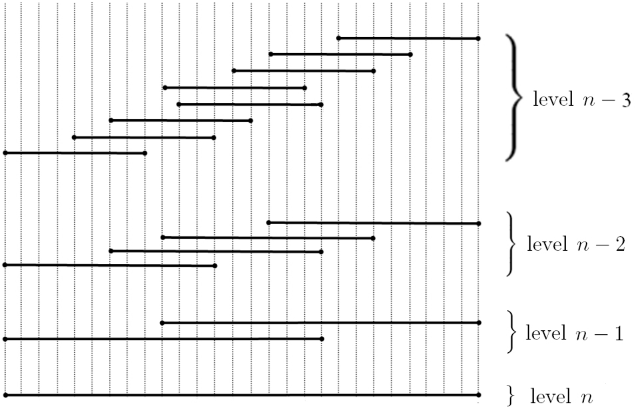
An in-depth example

We now walk through an in-depth example to demonstrate our algorithm.

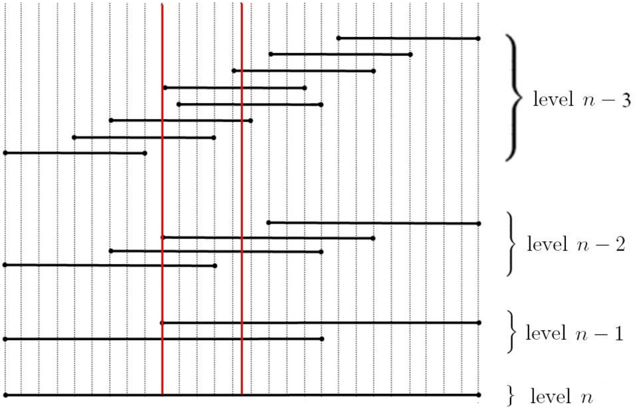
Each entry on level n can be written as a sum of entries of previous levels. In this particular example we write each entry on level n as a sum of entries on level $n-3$.

We break up the interval $[0,1]$ into subintervals of length $1/81$. Let's see what we get.

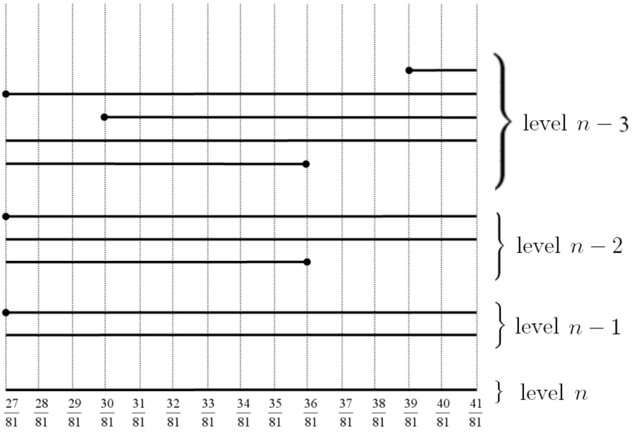
Pullback diagram



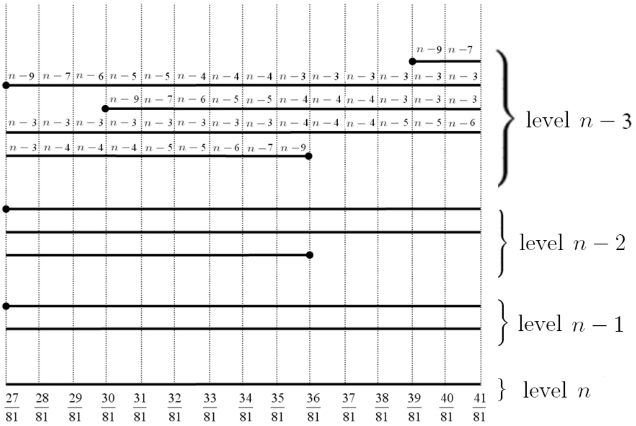
Pullback diagram



Pullback diagram



Pullback diagram



Largest real root

Interval	Polynomial	Largest real root
1	$x^n - 2\,x^{n-3} - x^{n-9}$	1.301688030
2	$x^n - x^{n-3} - x^{n-4} - x^{n-7}$	1.288452726
3	$x^n - x^{n-3} - x^{n-4} - x^{n-6}$	1.304077155
4	$x^n - x^{n-3} - x^{n-4} - x^{n-5} - x^{n-9}$	1.349240712
5	$x^n - x^{n-3} - x^{n-7} - 2\,x^{n-5}$	1.342242489
6	$x^n - x^{n-3} - x^{n-4} - x^{n-5} - x^{n-6}$	1.380277569
7	$x^n - x^{n-3} - x^{n-4} - x^{n-5} - x^{n-6}$	1.380277569
8	$x^n - x^{n-3} - x^{n-4} - x^{n-5} - x^{n-7}$	1.366811194
9	$x^n - x^{n-3} - 2\,x^{n-4} - x^{n-9}$	1.375394454
10	$x^n - x^{n-3} - 2\,x^{n-4}$	1.353209964
11	$x^n - x^{n-3} - 2\,x^{n-4}$	1.353209964
12	$x^n - 2\,x^{n-3} - x^{n-5}$	1.363964602
13	$x^n - 2\,x^{n-3} - x^{n-5} - x^{n-9}$	1.385877646
14	$x^n - 2\,x^{n-3} - x^{n-6} - x^{n-7}$	1.383834352

1.385877646

This example has given us the bound

$$m_n = O(1.385877646^n)$$

The actual proof uses induction on n. The details are rather involved; if anyone is interested I would be glad to go through the proof after the talk.

Data

The above example can be generalized to give an algorithm that computes a θ so that $m_n = (\theta^n)$.

Before implementing this algorithm, the best known bound on m_n was the $O((\sqrt{2})^n)$ bound given earlier. We succeeded in significantly improving the bound.

We have shown $m_n = O(\theta^n)$ where $\theta = 1.33997599527$. Specifically θ is a root of the polynomial

$$X^{33} - 752X^8 - 520X^7 - 319X^6 - 231x^5 - 141X^4 - 101X^3 - 54X^2 - 50X - 83$$

Conclusion

Bound

Previous Bound

$$O((1.41421356237)^n)$$

Our Bound

$$O((1.33997599527)^n)$$

Our Conjecture

$$O((1.\overline{333})^n)$$

Future Questions

These are questions we have been unable to answer:

Can our bound improvement algorithm be pushed to further lower the bound given more computational power?

Is it possible to conclusively prove our conjecture?

Furthermore, is there an explicit formula to describe m_n for any arbitrary level?

What else can be said regarding the global behavior of the Bernoulli sequence B_n?

However, we have provided partial answers for these questions through our algorithms and data.

Thank you for listening!