Shannon Theory and Magnetic Recording

Jack Keil Wolf

Shannon Day at Bell Labs

May 18, 1998

jwolf@ucsd.edu

http://cmrr-wolf04.ucsd.edu/wolf2/default.htm

Codes for Magnetic Recording

Two types of channel codes are used in magnetic recording:

- Error correction codes.
- Modulation (or constrained) codes.

Modulation Codes

Modulation codes serve several purposes:

- To help fight intersymbol interference.
- To aid in timing recovery.

One Dimensional Constraints

- The most common constraint previously used in recording is the (d,k) constraint.

- The code sequences are binary sequences where there is *at least* d and *at most* k 0's following every 1.

Encoder for (2,7) Constraint

Data Bits	Code Bits
10	0100
11	1000
000	000100
010	100100
011	001000
0010	00100100
0011	00001000

Rate for (2,7) 1-D Constraint

For the code shown previously number of code bits is always twice the number of data bits. The *rate* of the code is 1/2 or 0.500.

The *capacity* of the constraint is the *maximum rate* for any code that satisfies the constraint. The capacity for the (2,7) code is known to be 0.5174.

Definition of Capacity for the (d,k) Constraint

Let $N_{d,k}(n)$=the number of *distinct* sequences of code symbols of length n that satisfy the 1-D (d,k) constraint.

The capacity, $C_1(d,k)$, is equal to the limit as n approaches infinity of $\log_2(N_{d,k}(n))$ divided by n.

Shannon and the (d,k) Constraint

Shannon, in his classic 1948 papers, defined the *capacity of the noiseless channel* and gave several methods for computing this capacity.

This definition and the methods of computation apply to (d,k) constrained binary sequences.

Shannon and the (d,k) Constraint

Yet nowhere in these papers does Shannon discuss (d,k) constrained sequences nor for that matter any form of digital storage!

A Capacity Calculation

For k finite, Shannon showed by a simple counting argument that $N_{d,k}(n)$ satisfies the difference equation:

$$N_{d,k}(n) = N_{d,k}(n-(d+1)) + N_{d,k}(n-(d+2)) + \ldots + N_{d,k}(n-(k+1)).$$

By standard methods of solving difference equations, Shannon showed that the capacity is equal to the base 2 logarithm of the largest real root of the equation:

$$x^{(k+2)} - x^{(k+1)} - x^{(k-d+1)} + 1 = 0.$$

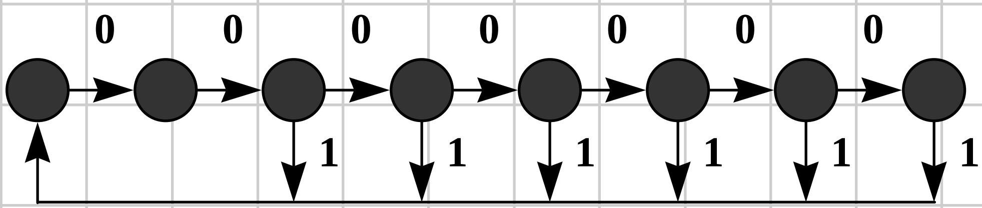
Shannon's Second Method

Shannon drew the constraint as a directed graph where the edges were labeled with the code symbols.

Shannon then showed that under certain conditions the capacity was equal to the base 2 logarithm of the largest eigenvalue of the *adjacency matrix* of the graph.

Second Method for 1-D (2,7) Constraint

The constraint graph looks like



and its adjacency matrix is:

$$
\begin{matrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
$$

Values of 1-D Capacity

	d=0	d=1	d=2
k=1	.6942		
k=2	.8792	.4057	
k=3	.9468	.5515	.2878
k=4	.9752	.6175	.4057
k=5	.9881	.6509	.4650
k=6	.9942	.6690	.4979
k=7	.9971	.6793	.5174

Proof That $C_1(1,2)=C_1(2,4)$

Take any (2,4) sequence such as:

1 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 1 . . .

Wherever there are a run of 3 or 4 0's, change the *second* 0 to a 1. This gives:

1 0 <u>1</u> 0 1 0 0 1 0 <u>1</u> 0 0 1 0 <u>1</u> 0 1 . . . which is a (1,2) sequence.

The process can be reversed to get a (2,4) sequence from a (1,2) sequence.

2-D Constrained Binary Arrays

These are constrained binary arrays where the binary digits are arranged in rows and columns.

Applications include: optical recording and multitrack magnetic and magneto optical recording. The application dictates the type of constraints.

2-D (d,k) Binary Arrays

Here we consider binary arrays where *every row* and *every column* satisfies the (d,k) constraint.

Shannon briefly mentioned two dimensional arrays with respect to constructing crossword puzzles.

A quote from Shannon (1948) follows:

Shannon and 2-D Crossword Puzzles

From Shannon 1948:

The ratio of the entropy of a source to the maximum value it could have while still restricted to the same symbols will be called its *relative entropy*. ... One minus the relative entropy is the *redundancy*.

Capacity of 2-D (d,k) Binary Arrays

As in 1-D, let $N_{d,k}(m,n)$ = the number of *distinct* arrays of code symbols with m rows and n columns that satisfy the 2-D (d,k) constraint.

The 2-D capacity, $C_2(d,k)$, is equal to the limit as m and n approach infinity of $\log_2(N_{d,k}(m,n))$ divided by (m)(n).

Computation of Capacity of 2-D (d,k) Binary Arrays

Shannon didn't tell us how to make this computation although his comment on crossword puzzles hints that the capacity of some 2-D arrays could be 0.

Almost nothing is known about computing the capacity for arbitrary choices of d and k.

The Capacity of 2-D (d,k) Binary Arrays

An exception to our lack of knowledge is that Kato and Zeger have shown that for every $d > 1$, $C_2(d,k)=0$ if and only if $k=d+1$.

Since $C_1(1,2) = C_1(2,4)$, both constraints have the same redundancy. Yet from Kato and Zeger:

$$C_2(1,2) = 0 \text{ but } C_2(2,4) > 0.$$

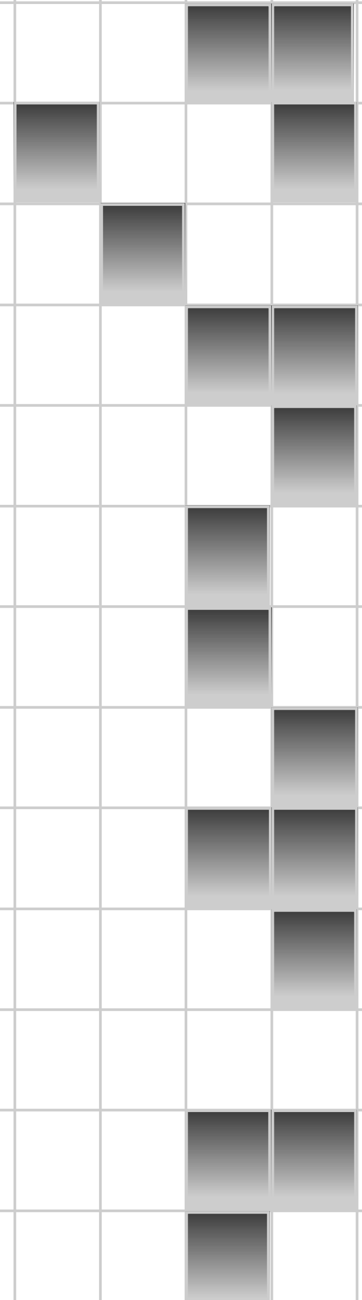
Proof That $C_2(1,2)=0$

One can start with any 1-D sequence satisfying the (1,2) constraint as the first row but then the remainder of the array is specified *except for one bit*.

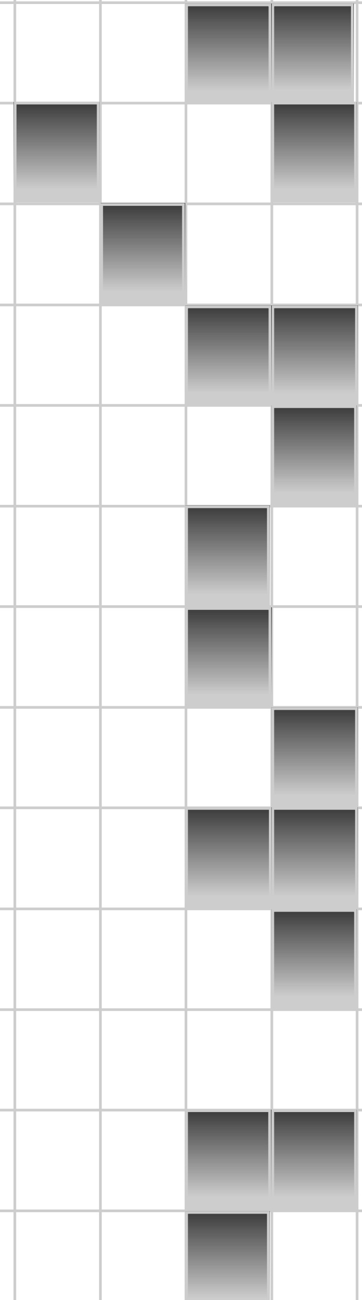
Example: Let the first row be:

0 1 0 1 0 0 1 0 0 1 0 1 0 0 1 . . .

0	1	0	1	0	0	1	0	0	1	0	1	0	0	1
	<u>0</u>	<u>1</u>	<u>0</u>	?	?	<u>0</u>			<u>0</u>	<u>1</u>	<u>0</u>			<u>0</u>

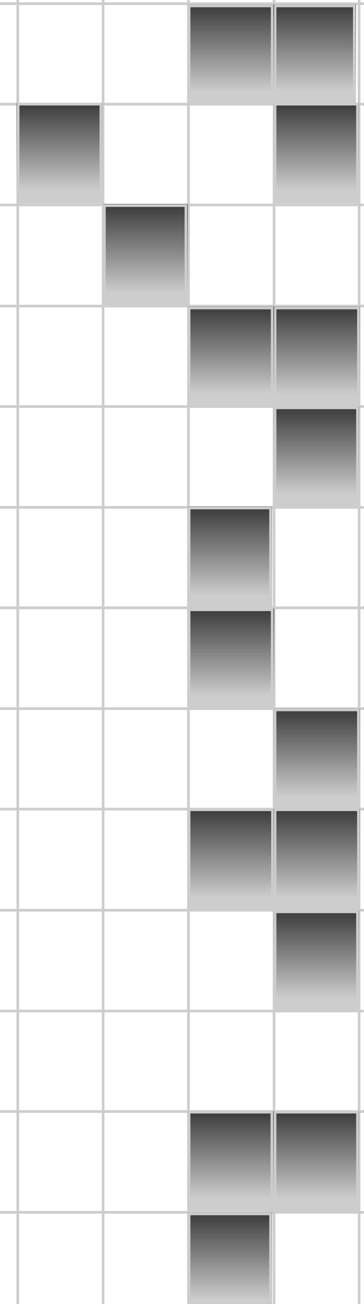


0	1	0	1	0	0	1	0	0	1	0	1	0	0	1
	0	1	0	<u>1</u>	<u>0</u>	0	?	?	0	1	0			0



0 1 0 1 0 0 1 0 0 1 0 1 0 0 1

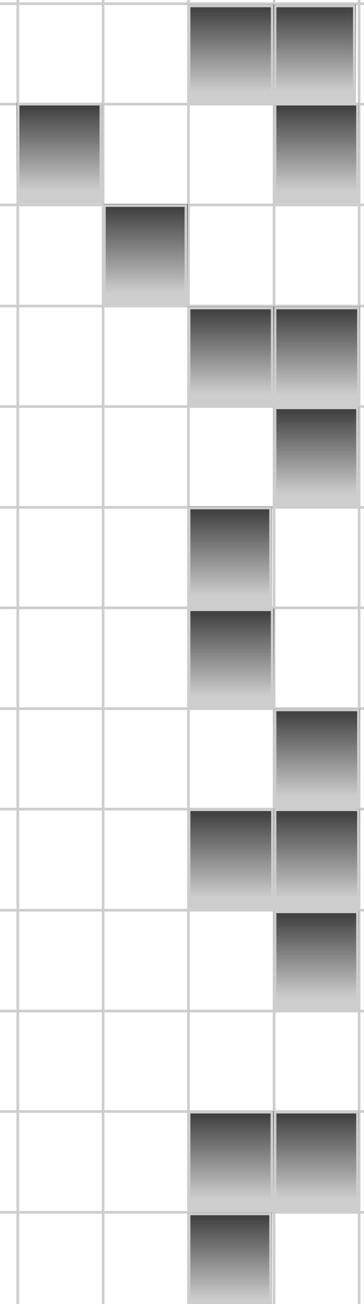
0 1 0 1 0 0 1 0 0 1 0 ? ? 0



0 1 0 1 0 0 1 0 0 1 0 1 0 0 1

0 1 0 1 0 0 1 0 0 1 0 ? ? 0

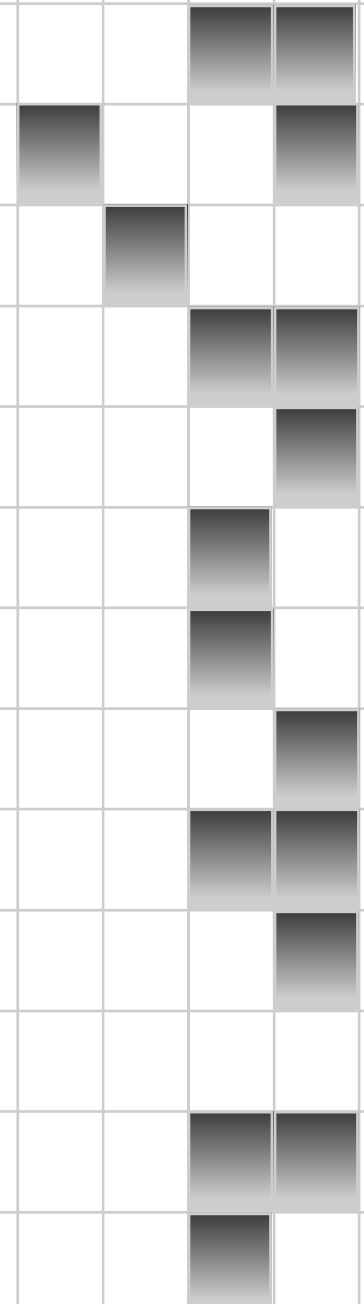
<u>1</u> <u>0</u> <u>0</u> <u>1</u> <u>0</u>



0 1 0 1 0 0 1 0 0 1 0 1 0 0 1

0 1 0 1 0 0 1 0 0 1 0 1 0 0

1 0 0 1 0



Bounds for C_2(d,inf)

One can use the methods developed by Shannon for computing the 1-D capacity to get bounds on the 2-D capacity.

Treat the N rows of the array as a *sequence* of N-vectors and determine its 1-D capacity which we call C_1(d,inf;N).

Bounds for $C_2(d,\inf)$

Then for every N $\geq$1

$C_1(d,\inf;N)/(N+d) < C_2(d,\inf) < C_1(d,\inf;N)/N$

The upper bound follows by choosing each set of N rows of the array from allowable sequences of these N-vectors and noting that the columns may violate the "d" constraint.

Bounds for $C_2(d,inf)$

The lower bound follows by inserting “d” rows of 0’s between every such set of N rows to insure that the columns satisfy the “d” constraint.

Bounds for $C_2(1,\text{inf})$

Example: Let N =2. The allowable 2-vectors are:

$$\begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 0. \end{matrix}$$

The all zero vector can be followed by any of the 3 vectors and the other two vectors can be followed by any vector but itself. The adjaceny matrix is:

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0. \end{matrix}$$

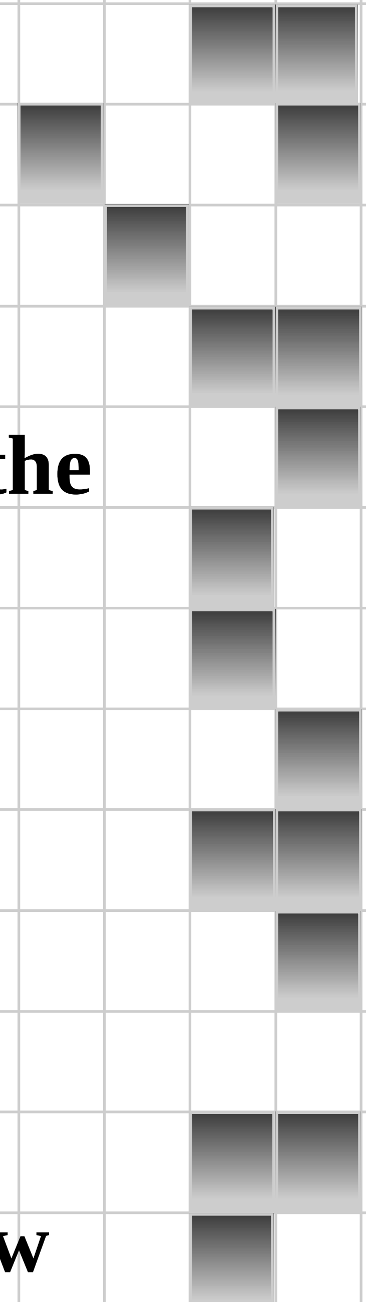
Bounds for $C_2(1,\inf)$

The base 2 logarithm of the largest real eigenvalue of this matrix is 1.2716.

An upper bound is thus 1.2716/2 = 0.6358.

A lower bound is thus 1.2716/3 = 0.4239.

Easier Lower Bounds for $C_2(1,\inf)$



$$C_2(1,\inf) \geq 0.5$$

since we can write uncoded data on the red squares and put 0's on the black squares (of a checkerboard).

$$C_2(1,\inf) \geq C_1(2,\inf) = 0.5515$$

since we can write a 1-D (2,inf) sequence down successive diagonals and place the two 0's that must follow every 1 to the right and below the 1.

Easy Lower Bound for $C_2(1,\inf)$

Sometimes, there is already a 0 to the right of the 1 (because the 1 was below the previous 1). In that case, just don't write that 0. This improves the lower bound.

In writing on a diagonal skip positions that contain a previously written 0.

Summary

Fred Jelinek captured the essence of my message in his illustration: