Information Theory Challenge #1

Andrew Steckley, PhD

Nov 1, 2021

1 The Problem

> What is the maximum amount of information that one can convey using a standard chess board and up to 5 pawns chosen from a set of 6 pawns that is evenly split by color?

Consider the problem illustrated in this way. Suppose you are a spy and you wish to communicate messages to another spy. But your only means of exchanging messages is by positioning a chess board and pawns on a table in a room that you both visit frequently, but are never in at the same time. And the pawns that you place on the board are chosen from a set of 3 white pawns and 3 black pawns. Fortunately, you have both been provided copies of a code book that lists N different possible messages. You just need to specify which one of the messages you want to convey to the other spy by configuring the board and pieces for them to see when they next visit the room. What is the highest value of N that this communication system can support?

This is not meant to be a trick-question in any way, so the above description should be fairly complete. But here are some additional details to avoid ambiguity:

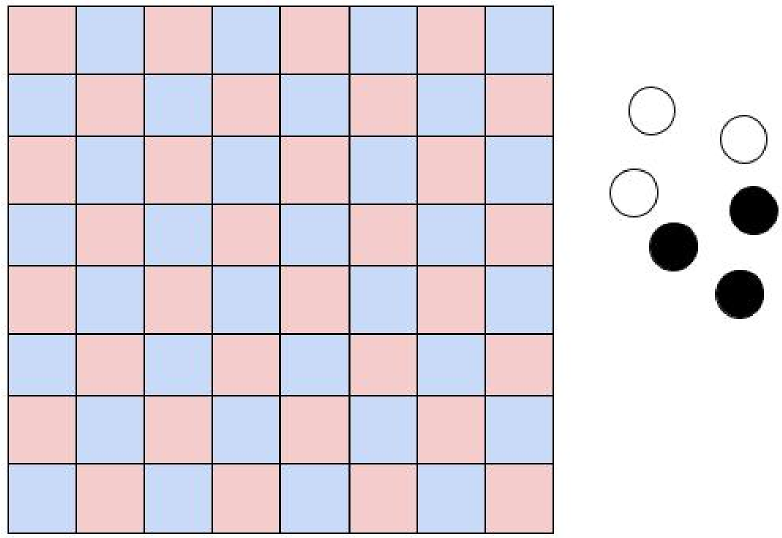
- It is a standard chessboard -- a simple 8x8 grid of alternately colored cells.
- The pawns are standard Staunton style pieces.
- The chessboard is always placed flat on the table. And a cell can only have one pawn on it at a time, placed upright in the center of the cell.
- The code book would include any instructions necessary to decode any particular configuration of the board and pawns into one of the available messages.

A couple of clues:

1. It’s more than you probably think it is.

2. It’s less than you probably think it is after adjusting what you thought it was for the fact that it is more than you probably think it is.

(Bonus points for estimating the weight of the code book!)



2 The Short Answer

$$N = 80,864,033 \quad \Longrightarrow \quad \text{26.27 bits of information}$$

But let's see how we get there...

3 The Solution Derived

The board is an 8 x 8 grid of alternately colored cells and as a result it is invariant under a rotation of 180^o. (Note this is a result of the number or ranks and files being equal and even. If the board were a square grid with an odd number of alternately colored cells, it would be invariant under rotations of 90^o and 270^o as well.)

First, let's consider the number or possible ways that pawns can be arranged on a chess board with a fixed orientation. Then we will consider the impact of the boards rotational symmetry.

For any selection of w white pawns and b black pawns, we can first place the white pawns onto the board in $\binom{64}{w}$ possible configurations. For each of these configurations, there will be $(64 - w)$ empty cells remaining on which to place the b black pawns and there would be $\binom{64-w}{b}$ ways to do this. So the total number of configurations would be $\binom{64}{w}\binom{64-w}{b}$.

We now need to sum up these numbers for each possible set of $\{w, b\}$. These values are limited by the fact that $w \leq 3$ and $b \leq 3$ and $(w + b) \leq 5$. The total number of configurations, which we will denote as $\hat{N}$, is then given by the following formula.

$$\hat{N} = \sum_{w=0}^{3} \left(\sum_{b=0}^{min(3,5-w)} \binom{64}{w} \binom{64-w}{b} \right)$$

We can generalize this formula as follows:

$$\hat{N} = \sum_{w=0}^{min(n_w,n)} \left(\sum_{b=0}^{min(n_b,n-w)} \binom{n_c}{w} \binom{n_c-w}{b} \right)$$

where

n_c is the total number of cells on the board

w is the number of white pawns in a configuration

b is the number of black pawns in a configuration

n is the maximum number of pawns in a configuration

n_w is the number of white pawns available from which to select

n_b is the number of black pawns available from which to select

$\binom{a}{b} = \frac{a!}{b!(a-b)!}$ represents the number of combinations obtained when choosing a elements from a set of b elements

This set of board-pawn configurations, however, does not yet represent the set of unique configurations. We must still consider the effects of the board's invariance under rotation.

Each configuration accounted for in the above set of $\hat{N}$, will be either symmetric or asymmetric. It will be asymmetric if, when rotated by 180^o, it is identical to one of the other configurations that has been counted in the total set. Consequently these two configurations are indistinguishable from each other and so can only be used to encode one possible message. If a configuration is not asymmetric then it is symmetric, in which case the rotated version is indistinguishable from the version prior to the rotation.

The symmetric configurations have been properly accounted for in the calculation of $\hat{N}$, but the asymmetric ones have been double counted. We need to account for the double counted asymmetric configurations. And to do this, we must determine the number or symmetric vs asymmetric configurations.

For a configuration to be symmetric, the pawns must be present in pairs. There must be an even number of white pawns and an even number or black pawns, and they must be arranged so that each pawn has a partner of equal color placed on the cell corresponding to a 180^o rotation of the board. We can count up these symmetric configurations by considering half the board and just one of the pawns from each pair of pawns. The formula becomes similar to the previous one, but we are now looking at 32 cells, and we must limit the numbers of pawns to be no more than half of the values previously considered.

This gives the number or symmetric configurations, which we will denote as $\tilde{N}$, as:

$$\tilde{N} = \sum_{w=0}^{min(\frac{n_w}{2},\frac{n}{2})} \left(\sum_{b=0}^{min(\frac{n_b}{2},\frac{n}{2}-w)} \binom{\frac{n_c}{2}}{b} \binom{\frac{n_c}{2}-b}{w} \right)$$

(Note that, since this is a counting formula, the truncation of fractional quantities to the nearest integer is implied. We have also now restricted the formula to apply to a board that is invariant under a rotation of 180^o. This implies that n_c must be the square of an odd integer.)

We now have the quantities needed to determine the total number of unique configurations -- that is the number or configurations that are distinguishable from each other regardless of the orientation of the board, and which therefore can each be used to encode a different message.

We just need to cut the number of asymmetric configurations, $(\hat{N} - \tilde{N})$, in half and then add the number or symmetric configurations:

$$N = \frac{\hat{N}-\tilde{N}}{2} + \tilde{N}$$

4 Python Code

In [1]:

```
import math
def c(n, r):
    return math.factorial(n)/(math.factorial(r)*math.factorial(n - r))
```

executed in 3ms, finished 08:21:26 2021-11-12

In [2]:

```
def a(n_c, n, n_b, n_w):
    print(f" Board-Pawn Configurations", flush=True)
    print(f"---------------------------", flush=True)
    total = 0
    for w in range(0, min(n_w, n) + 1):
        n_sum = 0
        for b in range(0, min(n_b, n - w) + 1):
            x = int(c(n_c, b)*c(n_c - b, w))
            n_sum += x
            print(f"    {b} b pawns, {w} w pawns ({(w + b)} pawns total):
        print()
        total += n_sum
    pass
    print(f"  Total unique configurations, non-orientable board = {total:,
    print(f"                                                     ==>  {math
    print()
    return total
```

executed in 7ms, finished 08:21:26 2021-11-12

```
In [3]: ef s(n_c, n, n_b, n_w, total_all=None ):
            print(f" Symmetric Board-Pawn Configurations", flush=True)
            print(f"-------------------------------------", flush=True)
            n_c2 = int(n_c / 2)
            total_symmetric = 0
            n2 = int(n / 2)
            nw2 = int(n_w / 2)
            nb2 = int(n_b / 2)

            for w in range(0, min(nw2, n2) + 1):
                for b in range(0, min(nb2, n2 - w) + 1):
                    x = int(c(n_c2, b)*c(n_c2 - b, w))
                    print(f"    {b} b pairs, {w} w pairs ({2*(w + b)} pawns total):
                    total_symmetric +=  x
                print()
            print(f"  Total number of symmetric configurations = {total_symmetric:,}'
            if a is not None:
                total_asymmetric = total_all - total_symmetric
                print(f"  Total number of asymmetric configurations = {total_asymmetr
            print()
            return total_symmetric, total_asymmetric
```

executed in 7ms, finished 08:21:26 2021-11-12

```
In [4]: def f(M_c, n, n_b, n_w):
            num_all = a(M_c, n, n_b, n_w)
            num_symmetric, num_asymmetric = s(M_c, n, n_b, n_w, num_all)
            grand_total_unique = int((num_all - num_symmetric)/2) + num_symmetric
            bits = math.log(grand_total_unique, 2)
            print(f" Unique Board-Pawn Configurations", flush=True)
            print(f"----------------------------------", flush=True)
            print(f"  Total unique configurations, orientable board = {grand_tota
            print(f"                                                 ==>  {math.lo
            return
```

executed in 6ms, finished 08:21:26 2021-11-12

5 The Answer

Now applying the above functions to calculate our case of an 8x8 grid of alternately colored cells with the selection of up to 5 pawns from a set of 3 white pawns and 3 black pawns, we get:

In [5]:
```
f(64, 5, 3, 3)
```
executed in 19ms, finished 08:21:26 2021-11-12

```
 Board-Pawn Configurations
---------------------------
    0 b pawns, 0 w pawns (0 pawns total):  (64 0) * (64 0) = 1
    1 b pawns, 0 w pawns (1 pawns total):  (64 1) * (63 0) = 64
    2 b pawns, 0 w pawns (2 pawns total):  (64 2) * (62 0) = 2016
    3 b pawns, 0 w pawns (3 pawns total):  (64 3) * (61 0) = 41664

    0 b pawns, 1 w pawns (1 pawns total):  (64 0) * (64 1) = 64
    1 b pawns, 1 w pawns (2 pawns total):  (64 1) * (63 1) = 4032
    2 b pawns, 1 w pawns (3 pawns total):  (64 2) * (62 1) = 124992
    3 b pawns, 1 w pawns (4 pawns total):  (64 3) * (61 1) = 2541504

    0 b pawns, 2 w pawns (2 pawns total):  (64 0) * (64 2) = 2016
    1 b pawns, 2 w pawns (3 pawns total):  (64 1) * (63 2) = 124992
    2 b pawns, 2 w pawns (4 pawns total):  (64 2) * (62 2) = 3812256
    3 b pawns, 2 w pawns (5 pawns total):  (64 3) * (61 2) = 76245120

    0 b pawns, 3 w pawns (3 pawns total):  (64 0) * (64 3) = 41664
    1 b pawns, 3 w pawns (4 pawns total):  (64 1) * (63 3) = 2541504
    2 b pawns, 3 w pawns (5 pawns total):  (64 2) * (62 3) = 76245120

  Total unique configurations, non-orientable board = 161,727,009
                                                     ==>  27.27 bits

 Symmetric Board-Pawn Configurations
-------------------------------------
    0 b pairs, 0 w pairs (0 pawns total):  (32 0) * (32 0) = 1
    1 b pairs, 0 w pairs (2 pawns total):  (32 0) * (32 1) = 32

    0 b pairs, 1 w pairs (2 pawns total):  (32 1) * (31 0) = 32
    1 b pairs, 1 w pairs (4 pawns total):  (32 1) * (31 1) = 992

  Total number of symmetric configurations = 1,057
  Total number of asymmetric configurations = 161,725,952

 Unique Board-Pawn Configurations
----------------------------------
  Total unique configurations, orientable board = 80,864,033
                                                 ==>  26.27 bits
```

So our answer is:

$$N = 80,864,033$$

or 26.27 bits of information. Hence the maximum number of messages our spies can exchange will be 80,864,033, which is plenty enough to confuse things and turn just about any critical mission into a FUBAR operation!

6 Some Other Example Cases

6.1 Generalization Example

As an illustration of the generalized formulae, what if we instead had a board that was a 10x10 grid of alternately colored cells and we were able to select up to 4 pawns from a set of 5 black pawns and 6 white pawns. This is given by:

In [6]:
```
f(100, 4, 5, 6)
```
executed in 20ms, finished 08:21:27 2021-11-12

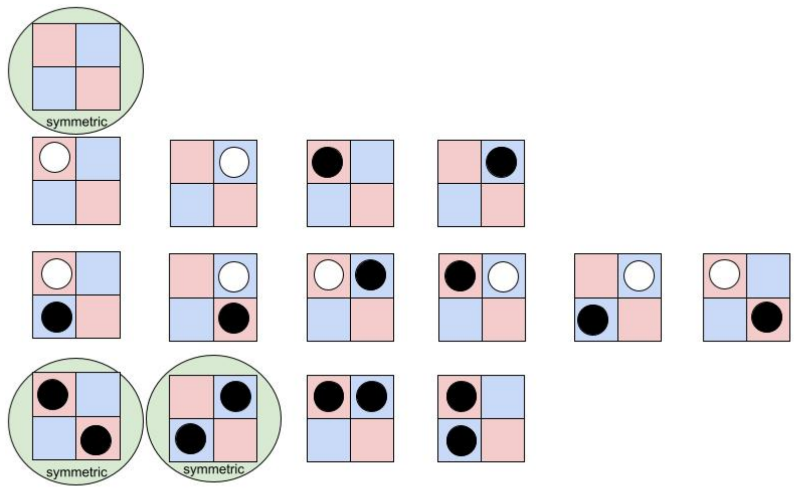
```
 Board-Pawn Configurations
---------------------------
    0 b pawns, 0 w pawns (0 pawns total):  (100 0) * (100 0) = 1
    1 b pawns, 0 w pawns (1 pawns total):  (100 1) * (99 0) = 100
    2 b pawns, 0 w pawns (2 pawns total):  (100 2) * (98 0) = 4950
    3 b pawns, 0 w pawns (3 pawns total):  (100 3) * (97 0) = 161700
    4 b pawns, 0 w pawns (4 pawns total):  (100 4) * (96 0) = 3921225

    0 b pawns, 1 w pawns (1 pawns total):  (100 0) * (100 1) = 100
    1 b pawns, 1 w pawns (2 pawns total):  (100 1) * (99 1) = 9900
    2 b pawns, 1 w pawns (3 pawns total):  (100 2) * (98 1) = 485100
    3 b pawns, 1 w pawns (4 pawns total):  (100 3) * (97 1) = 15684900

    0 b pawns, 2 w pawns (2 pawns total):  (100 0) * (100 2) = 4950
    1 b pawns, 2 w pawns (3 pawns total):  (100 1) * (99 2) = 485100
    2 b pawns, 2 w pawns (4 pawns total):  (100 2) * (98 2) = 23527350

    0 b pawns, 3 w pawns (3 pawns total):  (100 0) * (100 3) = 161700
    1 b pawns, 3 w pawns (4 pawns total):  (100 1) * (99 3) = 15684900

    0 b pawns, 4 w pawns (4 pawns total):  (100 0) * (100 4) = 3921225

  Total unique configurations, non-orientable board = 64,053,201
                                                     ==>  25.93 bits

 Symmetric Board-Pawn Configurations
-------------------------------------
    0 b pairs, 0 w pairs (0 pawns total):  (50 0) * (50 0) = 1
    1 b pairs, 0 w pairs (2 pawns total):  (50 0) * (50 1) = 50
    2 b pairs, 0 w pairs (4 pawns total):  (50 0) * (50 2) = 1225

    0 b pairs, 1 w pairs (2 pawns total):  (50 1) * (49 0) = 50
    1 b pairs, 1 w pairs (4 pawns total):  (50 1) * (49 1) = 2450

    0 b pairs, 2 w pairs (4 pawns total):  (50 2) * (48 0) = 1225

  Total number of symmetric configurations = 5,001
  Total number of asymmetric configurations = 64,048,200

 Unique Board-Pawn Configurations
----------------------------------
  Total unique configurations, orientable board = 32,029,101
                                                 ==>  24.93 bits
```

Even with a board that was more than 50% larger, we can support less than half the amount of information, because we have limited the number or pawns to less.

6.2 Hand-Checked Example

Finally, a much simpler case would be to use a small 2x2 board, and up to 2 pawns selected from a set of 2 black pawns and 1 white pawn. In this case we get just 15 possible configurations, which one can easily check out by hand.

The following diagram shows all 15 unique configurations. It also shows the 3 which are symmetric configurations. The other 12 cases are ones which, when rotated by 180^o produce a second configuration that is not actually distinguishable; these are ones which are therefore double-counted when simply counting up configurations for a fixed-orientation board.



In [7]:
```
f(4,2,2,1)
```
executed in 12ms, finished 08:21:27 2021-11-12

```
 Board-Pawn Configurations
---------------------------
    0 b pawns, 0 w pawns (0 pawns total):  (4 0) * (4 0) = 1
    1 b pawns, 0 w pawns (1 pawns total):  (4 1) * (3 0) = 4
    2 b pawns, 0 w pawns (2 pawns total):  (4 2) * (2 0) = 6

    0 b pawns, 1 w pawns (1 pawns total):  (4 0) * (4 1) = 4
    1 b pawns, 1 w pawns (2 pawns total):  (4 1) * (3 1) = 12

  Total unique configurations, non-orientable board = 27
                                                    ==>   4.75 bits

 Symmetric Board-Pawn Configurations
-------------------------------------
    0 b pairs, 0 w pairs (0 pawns total):  (2 0) * (2 0) = 1
    1 b pairs, 0 w pairs (2 pawn total):  (2 0) * (2 1) = 2

  Total number of symmetric configurations = 3
  Total number of asymmetric configurations = 24

 Unique Board-Pawn Configurations
----------------------------------
  Total unique configurations, orientable board = 15
                                                ==>   3.91 bits
```