An Assortment of Hat Puzzles

David C. Ross

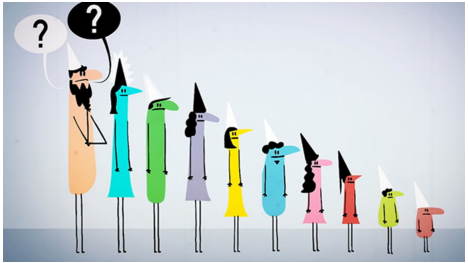
Student Seminar, Bar Ilan University

January 17, 2022

Table of Contents

Introduction

The prisoner hat riddle is one of the most classic induction puzzles. It can be phrased as: *Given n prisoners, every prisoner can see the hats of the people in front but not their own hat, or the hats worn by anyone behind. A prison guard, starts at the back of the line and asks each prisoner the colour of their hat. If they answer correctly, they will be pardoned, if they get it wrong, they'll be executed :).*



Introduction

Solution:

The prisoners come up with the following plan: If the first prisoner to speak—the one in the back of the line—sees an even number of blue hats, he will yell out "blue." If he sees and odd number of blue hats, he will yell out "red." From this the remaining prisoners are able to deduce the color of their own hats.

A Slightly More Interesting Problem

As before, the color of each player's hat will be determined by a fair coin flip. The players will be arranged in a circle so that each player can see all the other players' hats, and no communication will be permitted. Each player will then be taken aside and given the option of trying to guess whether his own hat is red or blue, but he may choose to pass.

The outcome is drastic: Unless at least one player guesses, and every player who does guess guesses correctly, all the players will be executed. It sounds like the best plan must be to have only one player guess and the rest pass; that, at least, gives them a 50% chance of survival. But, incredibly, the team can do much better— with 16 players, for example, they can improve their odds to better than 90%.

Building Intuition

First of all, let us try the game with three players. With three players, each player is instructed to pass if the two hats he sees are of different colors, otherwise to guess that his own hat is the color he does not see. The result is that as long as both colors are represented (as in six of the eight possible configurations), the odd player will guess correctly and the other two will pass. Thus, the players win with probability 75%.

Building Intuition

Notice that in the bad cases, where all three hats are the same color, *all* the players guess and they're all wrong. This is a crucial feature: The strategy packs six wrong guesses into only two configurations.

Over all configurations, half the guesses must be wrong, thus the only way to win is to use the correct guesses efficiently and cram the wrong guesses together.

The Idea

For n players, we would ideally like to duplicate this feat and have just the two kinds of configurations: "good" ones where just one player guesses (correctly) and "bad" ones where everyone guesses and they are all wrong. This would force good configurations to outnumber bad ones by a factor of n, giving us a gratifying winning probability of $\frac{n}{n+1}$.

Solution

We have no chance of achieving this perfect optimum unless $n + 1$ evenly divides the total number of configurations 2^n, which means that n must itself be one less than some power of 2. Somewhat miraculously, this condition is sufficient as well as necessary.

Solution

The key is to find a set of bad configurations that has the property that every other configuration is adjacent to exactly one bad one (adjacent means that you can get from one to the other by changing one hat color). Here is a way to define such a set.
Suppose $n = 2^k - 1$. Assign each player a different k-digit non-zero binary number (e.g., if there are 15 players, they get labeled 0001, 0010, $\ldots$, 1111). These labels are treated as "nimbers" not numbers: You can add them binarly without carry (i.e., xor), so e.g., $1011 + 1101 = 0110$ and anything plus itself is 0000.

Solution

The bad configurations are going to be the ones with the property that if you add all the labels of the red-hatted players, you get 0000.
The strategy is this: Each player adds all the nimbers belonging to the people he sees whose hats are red. If the sum is 0000, he guesses that his own hat is also red. If the sum is his own nimber, he guesses that his own hat is blue. If the sum is anything else, he passes.

But Why

So why in the world should this work? Well suppose the sum of the nimbers of *all* the people in red hats is, in fact 0000, Then everyone with a blue hat on will compute 0000 as the red hat sum, and will guess "red"; everyone with a red hat on will compute his own nimber as the sum and will guess "blue". Thus, every player will guess and every one of them will be wrong - just what we want!

But Why

Now suppose that the sum of the red-hat nimbers is something else, say 0101. Then the *only* player who guesses is the one whose nimber is 0101 and his guess with be *right*.

The probability that the red-hat nimbers sum is 0000 is exactly $\frac{1}{16}$. Thus, the strategy wins with the probability $\frac{15}{16}$; in general with probability $1 - 2^{-k}$. (It's easy to check that with nimbers of length 2, you get the three player solution back.)

If *n* does not happen to be one short of a power of 2, the easiest thing for the player to do is to compute the largest $m < n$ which is of the form $2^k - 1$. Those *m* players as above and the rest refuse to guess regardless of what they see. At worst (if $n = 2^k - 2$ for some integer k), This results in a winning probability of $\frac{n/2}{n/2+1}$. These strategies are not generally best possible; for $n = 4$ you can't beat 75%, but for $n = 5$ you can get $\frac{25}{32} > 78\%$. The best strategy for a general *n* is an unsolved problem.

But Why

Now suppose that the sum of the red-hat nimbers is something else, say 0101. Then the *only* player who guesses is the one whose nimber is 0101 and his guess with be *right*.

The probability that the red-hat nimbers sum is 0000 is exactly $\frac{1}{16}$. Thus, the strategy wins with the probability $\frac{15}{16}$; in general with probability $1 - 2^{-k}$. (It's easy to check that with nimbers of length 2, you get the three player solution back.)

If n does not happen to be one short of a power of 2, the easiest thing for the player to do is to compute the largest $m < n$ which is of the form $2^k - 1$. Those m players as above and the rest refuse to guess regardless of what they see. At worst (if $n = 2^k - 2$ for some integer k), This results in a winning probability of $\frac{n/2}{n/2+1}$. These strategies are not generally best possible; for $n = 4$ you can't beat 75%, but for $n = 5$ you can get $\frac{25}{32} > 78\%$. The best strategy for a general n is an unsolved problem.

But Why

Now suppose that the sum of the red-hat nimbers is something else, say 0101. Then the *only* player who guesses is the one whose nimber is 0101 and his guess with be *right*.

The probability that the red-hat nimbers sum is 0000 is exactly $\frac{1}{16}$. Thus, the strategy wins with the probability $\frac{15}{16}$; in general with probability $1 - 2^{-k}$. (It's easy to check that with nimbers of length 2, you get the three player solution back.)

If *n* does not happen to be one short of a power of 2, the easiest thing for the player to do is to compute the largest $m < n$ which is of the form $2^k - 1$. Those *m* players as above and the rest refuse to guess regardless of what they see. At worst (if $n = 2^k - 2$ for some integer k), This results in a winning probability of $\frac{n/2}{n/2+1}$. These strategies are not generally best possible; for $n = 4$ you can't beat 75%, but for $n = 5$ you can get $\frac{25}{32} > 78\%$. The best strategy for a general *n* is an unsolved problem.

Hamming Code

The set of bad configurations we constructed above is not only a beautiful mathematical object, but one which is useful in real life. It is called a Hamming code and is an example of a perfect *error-correcting code*.

Imagine that you are sending binary information over an unreliable channel which occasionally flips a bit. Group the bits you want to send into strings of size 11. There are $\frac{2^{15}}{16} = 2^{11}$ red-blue sequences of length 15 with the property that the sum of the red-hat nimbers is 0000. These special strings, which you can write in binary (101010101010101 means that every odd hat is red), are called "codewords"; since the number of codewords is 2^{11}, you can associate one with every 11-bit binary string. One easy way to do this is to chop off the last 4 bits. Now instead of sending your 11-bit group, you send the unique associated 15-bit codeword. You pay a price in efficiency, but you get something back; reliability. This is because if one of the 15 bits is accidentally flipped, the person who gets your message can tell which one and flip it back!

Hamming Code

The set of bad configurations we constructed above is not only a beautiful mathematical object, but one which is useful in real life. It is called a Hamming code and is an example of a perfect *error-correcting code*.

Imagine that you are sending binary information over an unreliable channel which occasionally flips a bit. Group the bits you want to send into strings of size 11. There are $\frac{2^{15}}{16} = 2^{11}$ red-blue sequences of length 15 with the property that the sum of the red-hat nimbers is 0000. These special strings, which you can write in binary (101010101010101 means that every odd hat is red), are called "codewords"; since the number of codewords is 2^{11}, you can associate one with every 11-bit binary string. One easy way to do this is to chop off the last 4 bits. Now instead of sending your 11-bit group, you send the unique associated 15-bit codeword. You pay a price in efficiency, but you get something back; reliability. This is because if one of the 15 bits is accidentally flipped, the person who gets your message can tell which one and flip it back!

Hamming Code

How? When he receives the 15 bits, he can add up the red nimbers (those corresponding to ones in the sequence) and make sure they add up to 0000. Suppose they don't, e.g., they add up to 0101. So a bit must have been flipped; if it was only one bit, it must have been the fifth bit. So the receiver unflips the fifth bit and checks his codebook to see which 11-bit sequence corresponds to the 15-bit codeword you intended to send. He will be right unless multiple bits were flipped.

Table of Contents

Framing the Problem

Seven prisoners are given the chance to be set free tomorrow. An executioner will put a hat on each prisoner's head. Each hat can be one of the seven colors of the rainbow and the hat colors are assigned completely at the executioner's discretion. Every prisoner can see the hat colors of the other six prisoners, but not his own. They cannot communicate with others in any form, or else they are immediately executed. Then each prisoner writes down his guess of his own hat color. If at least one prisoner correctly guesses the color of his hat, they all will be set free immediately; otherwise they will be executed. They are given the night to come up with a strategy. Is there a strategy that they can guarantee that they will be set free?

Classic Solution

The prisoners number the colors 0 through 6; say red $= 0$, orange $= 1$, yellow $= 2$, and so on through violet $= 6$. They also number themselves 0 through 6; We'll simply call them $P_0, P_1, \ldots, P_6$. When the hats are put onto their heads, each prisoner performs the following calculation: he adds up the numbers of the colors of the six hats that he can see and subtracts that from his own personal number. For example, if P_3 sees hats with colors $0, 2, 2, 5, 5$, and 6, he calculates

$$3 - (0 + 2 + 2 + 5 + 5 + 6) = -17$$

Then he reduces this modulo 7 to a number in the range from 0 through 6. In this case $-17 + 3 \cdot 7 = 4$ is the final result. This is the number corresponding to the color blue, so he writes down *blue*. The claim is that if every prisoner follows this procedure, one will write down the name of the color of his own hat.

Classic Solution

The prisoners number the colors 0 through 6; say red $= 0$, orange $= 1$, yellow $= 2$, and so on through violet $= 6$. They also number themselves 0 through 6; We'll simply call them $P_0, P_1, \ldots, P_6$. When the hats are put onto their heads, each prisoner performs the following calculation: he adds up the numbers of the colors of the six hats that he can see and subtracts that from his own personal number. For example, if P_3 sees hats with colors $0, 2, 2, 5, 5$, and 6, he calculates

$$3 - (0 + 2 + 2 + 5 + 5 + 6) = -17$$

Then he reduces this modulo 7 to a number in the range from 0 through 6. In this case $-17 + 3 \cdot 7 = 4$ is the final result. This is the number corresponding to the color blue, so he writes down *blue*. The claim is that if every prisoner follows this procedure, one will write down the name of the color of his own hat.

But Why

Here's why it works. Note first what happens if P_3's hat really is blue: then the hat colors add up to

$$0 + 2 + 2 + 5 + 5 + 6 + 4 = 24,$$

and when we reduce this modulo 7 we get $243 \cdot 7 = 3$,P_3's personal number. The procedure ensures that this happens with each prisoner: he guesses the hat color that would make the sum of all seven hat colors equal (modulo 7) to his own personal number.

Whatever the actual colors of the hats, their numbers must add up (modulo 7) to one and only one of the seven numbers $0, 1, 2, 3, 4, 5$, or 6. Say they add up to k. Then P_k and only P_k writes down the color that makes the total correct. Every other prisoner writes down a color corresponding to one of the other six possible totals.

But Why

Here's why it works. Note first what happens if P_3's hat really is blue: then the hat colors add up to

$$0 + 2 + 2 + 5 + 5 + 6 + 4 = 24,$$

and when we reduce this modulo 7 we get $243 \cdot 7 = 3$,P_3's personal number. The procedure ensures that this happens with each prisoner: he guesses the hat color that would make the sum of all seven hat colors equal (modulo 7) to his own personal number.
Whatever the actual colors of the hats, their numbers must add up (modulo 7) to one and only one of the seven numbers $0, 1, 2, 3, 4, 5$, or 6. Say they add up to k. Then P_k and only P_k writes down the color that makes the total correct. Every other prisoner writes down a color corresponding to one of the other six possible totals.

But Why

Let's say that P_k wrote down color number c_1, that his own hat's color is number c_2, and that the color numbers of the six hats that he could see added up to t. Then on the one hand the procedure ensures that he chose c_1 to make $t + c_1$ equal to his own number, k, modulo 7, and on the other hand we know that k is the actual total of the seven color numbers, so $t + c_2$ is equal to k modulo 7. In short, $t + c_1$ and $t + c_2$ reduce to the same number modulo 7, so $c_1 = c_2$: he wrote down the color of his own hat.

The End