1 HAT COLORS AND HAMMING CODES

1.1 THE GAME

Number of players:	3, 7, or 15
You will need:	2 colored hats or caps for each player (1 in blue and 1 in red); alternatively, colored tags that can be attached to the hats

The players sit in a room, wearing red and blue hats. Each player's hat color has been chosen by members of the audience. Every player is able to see all the hats except their own. Below is an example of Anwar, Bella, and Charlie sitting in a circle, with Anwar wearing a red hat and Bella and Charlie wearing blue ones. The players are not allowed to talk to one another, and after a few seconds, they have to hold up a sign with their answers to the question: "What is the color of your hat?" The answers are "red," "blue," or "?" (which corresponds to "I don't know"). All players win or lose together. They win the game if and only if at least one of the answers is correct and no player gives a wrong answer. Question marks never count as incorrect, but wearing a blue hat and answering "red" is wrong, and so is answering "blue" when wearing a red hat. If, for example,

Bella and Charlie correctly answer "blue" and Anwar says "I don't know," they have won the game. But if all 3 players answer "blue," they lose (because Anwar's answer is incorrect).

What is the best strategy? How likely is it that the group of three players will win the game? What if there were n *players?*

Let us first assume that the audience distributes the hats at random, that is, the color of one player's hat does not contain any information about the color of any of the other player's hat, and each color is chosen with the same probability. It is perhaps surprising that there is an interesting strategy at all!

The group of players could, of course, decide on one person (e.g., Anwar) to always guess "blue." If the others answer "I don't know," the probability of winning as a group is 1/2. Anwar can also choose "red" or "blue" at random, but his chance of getting the color correct would still be just 1/2, so the group will still only win half of all games on average. The group could decide to let more of the players guess their color, either at random or by picking a color beforehand. The chance of winning in this way would be just 1/4 if two players are allowed guess the color. Both players announcing a color would need to be correct, and each

of them is correct with probability 1/2. The chance that they are both correct is then the product of the individual probabilities, which is 1/4. This drops to just 1/8 if three players are allowed to pick their colors. Letting just one player pick his or her color at random is better than letting more people pick their colors at random.

But what else could Anwar do instead of just picking the color at random or in a predefined way? Say that he bases his decision on the colors of Bella's and Charlie's hats. These colors do not contain any information about the color of Anwar's hat, as his color was chosen independently of the other colors. In other words, regardless of any observed sequence of the colors of Bella's and Charlie's hats, the color of Anwar's hat will still be blue or red with equal probability.

We encourage the reader to pause here and think again about the questions presented on the previous page.

1.2 HOW WELL CAN A STRATEGY WORK?

Individually, we have no predictive power, and any guess of an individual's hat color is equally likely to be correct or wrong. Therefore, the key to success will be to "collect" the wrong answers in single instances of the game. That is, the players need to ensure that either just one person is correct (and the others say they do not know) or that all players announce their colors falsely, thus effectively bundling the false guesses into the same game and spreading out the correct guesses over as many games as possible.

We are now going to introduce some notation that will later help us to understand the general solution. We first associate

$$\text{blue with a } 0 \quad \text{and} \quad \text{red with a } 1.$$

In the example, the colors (red, blue, blue) for players 1, 2, and 3 (Anwar, Bella, and Charlie, respectively) are then equivalent to

the vector $(1, 0, 0)$. In the following, we will simply write (100) and call this a sequence rather than a vector. This notation works, too, of course, if there are n players in total. The players' hat colors can then be described by a sequence with n entries. These sequences look like (0110100011), for example, if $n = 10$.

If the players choose a deterministic strategy, their answers will always be identical if they see a specific sequence of colors on the other players. For each true sequence x of colors, we then get one collection of answers for Anwar, Bella, and Charlie. These answers can be represented in a matrix.[1] Its rows correspond to the 8 different color sequences x (each occurring with probability 1/8), and each column shows the answers given by one of the three players. Any strategy the players adopt will lead to outcomes that can be represented as follows:

		answers of			
true sequence		Anwar	Bella	Charlie	group win or loss
$x = (000)$		correct	no answer	correct	win
$x = (001)$		no answer	correct	wrong	loss
$x = (010)$	$\rightarrow$	no answer	no answer	wrong	$\rightarrow$ loss
$x = (100)$		wrong	wrong	correct	loss
$\vdots$			$\vdots$		$\vdots$

Here, only the first row corresponds to a successful run, as at least one player has answered correctly and nobody gave a wrong answer. In the example, Anwar has decided to guess his color only if the colors of Bella's and Charlie's hats agree. He then goes for the same color. In contrast, Bella copies Anwar's color but only if Charlie does not also have the same color. And Charlie always copies Bella's color.

In the first game—that is, the game represented by the first row—both Bella and Charlie have a blue hat on, and Anwar

1. A matrix is a table with rows and columns. See appendix B.7: "What Is … a Matrix?"

guesses "blue," which happens to be right. In the fourth game, Bella and Charlie also have blue hats on, but Anwar's guess turns out to be wrong. He decides not to provide an answer in the second and third games, as Bella's and Charlie's colors do not agree. From Anwar's point of view, the first and fourth game settings look identical. If he decides to guess a color (rather than saying "I don't know"), one of these game settings will yield a correct answer and the other row will yield a wrong answer. Thus for each player (i.e., for each column of the matrix above), the number of right answers equals the number of wrong answers, which again shows that, working as individuals, the players have no predictive power. If adopting a randomized strategy, each individual player will be no more likely to make a correct guess than a wrong one.

If this constraint (i.e., that there are the same number of right and wrong answers in each column) were the only constraint, how well could we do? The key is to bundle all the wrong answers into the same row and spread out the correct answers among as many rows as possible. A good arrangement would, for example, be the following:

		answers of			
true sequence		Anwar	Bella	Charlie	group win or loss
$x = (000)$		wrong	wrong	wrong	loss
$x = (001)$		no answer	no answer	correct	win
$x = (010)$	→	no answer	correct	no answer	→ win
$x = (100)$		correct	no answer	no answer	win
⋮		⋮	⋮	⋮	⋮

Here, all players guess their hat color incorrectly in case of the first color sequence (000) but then spread out their correct answers among the other color sequences (along with "I don't know" answers from other players), thereby maximizing the number of games they win. For any row with a wrong answer, we get, at best, three rows with a correct answer. Therefore, the group has to lose in at least one in four equally likely color

sequences (since rows without any answer are lost, too). So for any strategy,

$$P(\text{group loses}) \geq \frac{1}{4}.$$

With n instead of three players, the same argument yields

$$P(\text{group loses}) \geq \frac{1}{n+1}.$$

To obtain this bound, we have only used the fact that the hat color of a given player is independent of all the other hat colors, which led to the constraint of equal numbers of right and wrong answers for each player. Can we find a strategy for which the probability of losing comes close to this bound?

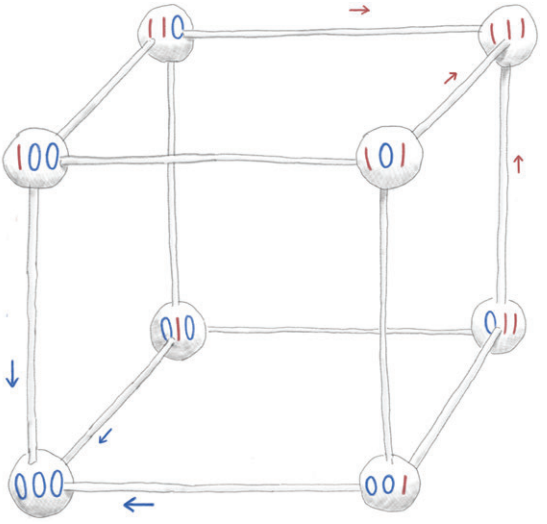
1.3 SOME MATHEMATICS: HAMMING CODES

Let us suppose that we would like to send a message from a sender to a receiver over a noisy channel, as if we were making a call on our cell phone, which is being disrupted by atmospheric turbulence and flocks of birds. Suppose we only want to send a sequence of letters $\{a, b\}$ (e.g., the sequence *abba*). In digital systems, we translate this message into a binary sequence containing only 1s and 0s. This step is called "encoding," and it works for many practical applications: We can then send the message using a digital system with 0 and 1 being low and high voltage levels, respectively, for example. However, most physical systems are noisy, and we can now try to find a binary code that will enable us to detect the noise in the transmission and remove it, at least partially.

Suppose we encode a as (000) and b as (111). Then the transmission of a sequence *abba* would look like

$$abba \overset{\text{encoding}}{\longrightarrow} 000\ 111\ 111\ 000 \overset{\text{noisy channel}}{\longrightarrow} 010\ 111\ 101\ 001 \overset{\text{decoding}}{\longrightarrow} abba.$$

The noise in the channel is flipping some of the 0s into 1s and vice versa. The first *a* is received as a sequence (010) by the receiver. It can either be decoded as *a* or *b*. The two codewords used are (000) for *a* and (111) for *b*. The sequence (010) can originate from (000) in a single flip. Starting from (111), we would need at least two flips. In this sense, the received (010) is closer to the codeword used for *a* than for *b*. Here, the distance is measured as the so-called Hamming distance, which simply counts the number of positions, or "bits," in which the received sequence and the codewords differ.[2]



2. The length of codewords is measured in "bits." We say that the encoded message on the previous page has 12 bits. Bits also appear in chapter 5, where they are used for measuring the information content conveyed by the outcome of a random variable.

The cube on the previous page illustrates the decoding. Suppose that we send either a or b. The collection of codewords (also called the "code") contains (000) for a and (111) for b. Each node corresponds to a message that could be received. The nodes (000), (100), (010), and (001) map to the codeword (000) for a, whereas all other nodes map to the codeword (111) for b. If the codewords are transmitted with a maximum error of a single flip, then the decoded sequences will match the true sequences perfectly. In this example, the coding scheme is said to be 1-*error correcting*, as a single flip in the transmission leads to no mistakes. It is also called *perfect*, as all sequences are either codewords themselves or can be generated by a codeword through a single flip. This ensures that after receiving a message, we are never in any doubt as to which codeword to choose in the decoding step. Codes that are 1-error correcting and perfect are called *Hamming codes*.

Decoding the sequence corresponds to associating the received sequence with the closest codeword. If a codeword is disrupted by at most one flip of a bit, then the decoded letter will be identical to the original letter. A mistake will happen if two or all three bits have been flipped by the noise in the channel. If each bit is flipped by the noisy channel with probability $p \in [0, 1/2]$, then each bit is not flipped with probability $(1-p)$. And therefore, each letter is recovered with probability[3]

$$(1-p)^3 + 3(1-p)^2 p.$$

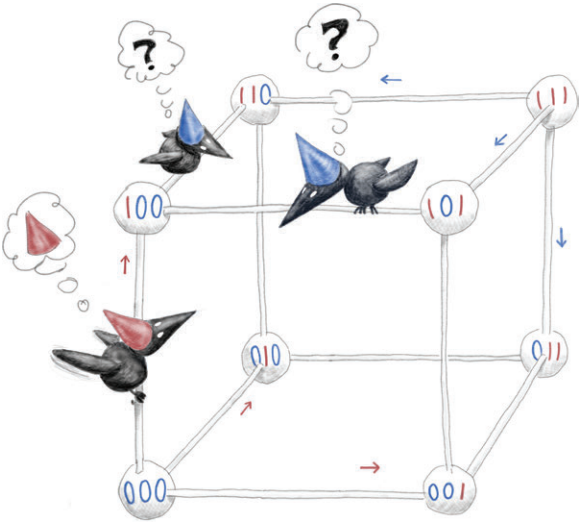
The term $(1-p)^3$ is the probability that the sequence has no flips. The term $3(1-p)^2 p$ is the probability of having only a single flip in the sequence: The probability that only the first bit is flipped is $(1-p)^2 p$, which is also the probability that only the second bit is flipped, and the same for the third. Taking the two

3. The factor 3 in the second term is the binomial coefficient $\binom{3}{1}$; see appendix B.4.

terms together, we obtain the probability of having no flip or just a single flip in the sequence (and hence not making a mistake, as both of these cases will be decoded correctly). The probability of not making a mistake is 97.2% for $p = 0.1$, compared with a probability of 90% of receiving the true sequence if we encode the letter a as 1 and transmit b as 0.

1.4 SOLUTION

Let us use a strategy based on the Hamming code that we discussed in Section 1.3. To see how this works, arrange the color sequences on a cube in three dimensions (or n dimensions for n players). All eight possible color sequences of a three-person game correspond to a corner of a cube.

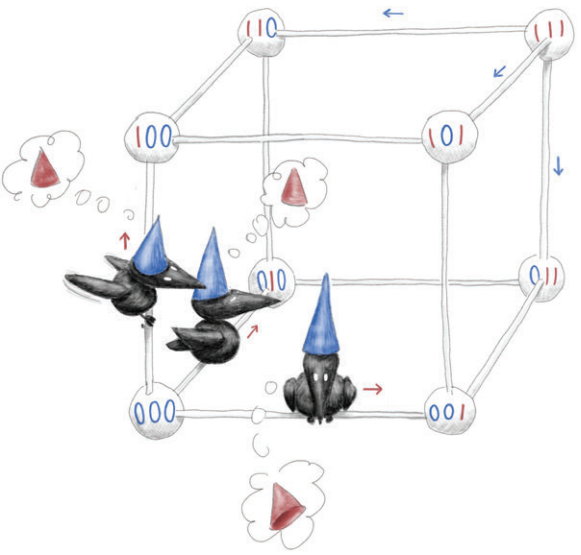


If the true sequence is (100), then the game "sits" at the top-left-front corner of the cube, as shown above. Anwar cannot see

that the game is in state (100), of course, as he cannot see the color of his own hat and only knows that the state is either (100) or (000), depending on whether his hat is red or blue. Therefore, we can say that Anwar sits on the edge on the front left, because, for Anwar, this edge connects the two sequences (100) and (000) that are compatible with his observations. Likewise, Charlie can be seen to sit on the top-left edge and Bella on the top front. All edges of the players connect to the true sequence (100), because the true sequence is a possible sequence for all players.

Now, Anwar can decide either to move to the upper node (100) by saying that his hat is red (which it really is), or he can decide, mistakenly, that his hat is blue and thus move to the lower node (000). As a third option, he can decide not to do anything and to answer that he does not know.

Hamming codes allow us to define a successful strategy. If you are sitting on an edge without a codeword as a neighbor, you should say "I don't know." However, if you are sitting next to a codeword, you should choose the sequence on the other side (this is the opposite of the decoding step after transmission over a noisy channel, where you should move toward the codeword, not away from it). We will first illustrate this strategy for three dimensions, $n = 3$, where the codewords are (000) and (111). We will consider the case where $n = 2^m - 1$ dimensions for an arbitrary $m \in \mathbb{N}$ in section 1.5. For the previous example, Anwar (on the front-left edge) is following the arrow on the edge and arrives at the answer "I have a red hat," as the arrow is pointing toward (100) and away from the codeword (000). The other players do not have an arrow on their edges, and they say "I don't know." So, the group wins. In general, if the true sequence is not a codeword of the Hamming code, the group will win, as exactly one player will identify her color correctly, and the rest of the group will not guess a color at all.



The cube above shows the outcome for the true sequence (000), that is, all players have a blue hat. This sequence coincides with a codeword of the Hamming sequence. All players have an arrow on their edge, and all edges point the wrong way: They will all announce the wrong color (red). The group therefore loses if the true sequence is a codeword.

Therefore, the outcomes are exactly as envisaged when discussing the best possible bound of the failure probability of the group. Either one player produces the correct answer and the others say they do not know, or all players give a wrong answer. A perfect 1-error correcting code guarantees that all states are at most distance 1 from a codeword, and that all neighbors of a codeword are not neighbors of another codeword (the definition of a perfect 1-error correcting code will be explained in more detail in section 1.5). Using the above strategy, the probability of losing is therefore equal to the fraction of codewords among

all possible sequences—still assuming that all sequences occur with equal probability. In the example, we have two codewords among eight sequences and hence

$$P(\text{group loses}) = \frac{1}{4}.$$

In the general case of n players, we will see in section 1.5 that the loss probability equals

$$P(\text{group loses}) = \frac{1}{n+1}.$$

This matches the bounds we derived above, and therefore, the win probabilities cannot be improved by using any other strategy.

1.5 HAMMING CODES IN HIGHER DIMENSIONS

In the coding example above, we encoded $\{a, b\}$ by using the codewords $W = \{(000), (111)\}$. We called (000) and (111) sequences, but, more formally, they can be seen as vectors in $\{0, 1\}^3$. Why did we call the code W perfect and 1-error correcting?

The *Hamming distance* $d(x, y)$ for $x, y \in \{0, 1\}^n$ is defined as the number of entries in both vectors that disagree,

$$d(x, y) := \#\{k : x_k \neq y_k\},$$

for example, $d(000110, 010100) = 2$. A ball $B_e(x)$ around $x \in \{0, 1\}^n$ with radius $e > 0$ is the set of all points $y \in \{0, 1\}^n$ that have Hamming distance at most e from x, that is,

$$B_e(x) := \{y \in \{0, 1\}^n : d(x, y) \leq e\}.$$

A code with codewords W is now said to be *e-error correcting* if for all $x, x' \in W$ with $x \neq x'$ and all $y \in B_e(x)$, we have

$$d(x, y) < d(x', y).$$

In words, if y is formed by changing a codeword x in e entries, then y will still be closer to x than to any other codeword $x' \in W$. An e-error correcting code W is said to be *perfect* if

$$\bigcup_{x \in W} B_e(x) = \{0, 1\}^n,$$

that is, the union of all e-balls around codewords is the whole set of sequences $\{0, 1\}^n$. A perfect e-error correcting code is called a *Hamming code*.

The strategy described in the previous section relies on having access to a Hamming code. For $n = 3$ players, we showed that $W = \{(000), (111)\}$ can be used, but we could equally well have used $W = \{(110), (001)\}$, for example. Assume that we manage to construct a 1-error correcting perfect code for the general case of n players. Then, for each codeword, there are n sequences that have Hamming distance 1 to this codeword. As the code is perfect, every sequence is either a codeword or has distance 1 to a codeword. For every unsuccessful outcome of the game (the true sequence is a codeword), there are n successful outcomes (the true sequence is not a codeword), which means that the probability of losing is

$$P(\text{group loses}) = \frac{1}{n+1}.$$

The challenge is to construct a perfect 1-error correcting Hamming code for the dimension n that matches the number of players. Currently, this can only be solved when the number of players equals $n = 2^m - 1$ for some $m \in \mathbb{N}$, for example, $n \in \{3, 7, 15, 31\}$; appendix C.1 shows how such codes can be constructed.

Adversarial Audience

If hat colors are drawn uniformly at random, each of the two sequences all-blue or all-red (which lead to failures of the strategy) occur only with probability 2^{-n}. They might occur more

often in practice, however, since the audience might be curious to see what happens in these somewhat special cases. Even worse, if a mischievous audience knows about the Hamming codes, they can always chose a Hamming codeword as the distribution of hats; using the above strategy, this makes the group lose every game. The group, however, can easily protect itself against such opposition by adding a randomly chosen sequence $\{0,1\}^n$ to all of the codewords and thereby creating a new valid Hamming code (we explain this in more detail in appendix C.1). If the players generate a new Hamming code before each game, they can ensure that the error probability remains the same, no matter what strategy the audience members employ.

1.6 SHORT HISTORY

Todd Ebert introduced what he called the "hat problem" in his PhD thesis at the University of California at Santa Barbara in 1998 [Ebert, 1998]. Many mathematicians tried to solve this problem until Elwyn Berlekamp, a Berkeley math professor, discovered a connection to coding theory and constructed the optimal strategy for the special cases $n = 2^m - 1$ for $m \in \mathbb{N}$. The problem for arbitrary n is still unsolved. Prior to Ebert's game, a similar game was introduced in an article titled "The expressive power of voting polynomials" [Aspnes et al., 1994], in which players cannot take the "I don't know" option, and the objective is to make sure that the majority of players answers correctly. Both games are also described in the *Mathematics Intelligencer* [Buhler, 2002].

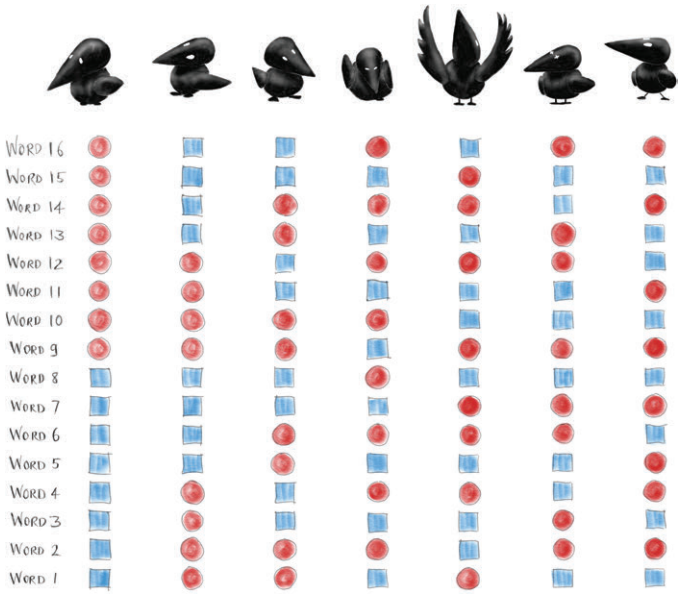
1.7 PRACTICAL ADVICE

The figure on the following page shows codewords for two 1-error correcting Hamming codes with $n = 3$ players. The rows in the left half of the figure correspond to the sequences (000) for all blue hats and (111) for all red hats.

One can add a constant to all codewords, as shown on the right. This yields a new valid Hamming code and can thus be used, too.

The figure above can now be used to derive the optimal action for each player. Suppose we are player 3 in a game with $n = 3$ players, and we use the codewords on the left in the figure (all red for word 1 and all blue for word 2). Then we check the colors of players 1 and 2. If they do not match any of the rows for players 1 and 2 in the list of codewords, we answer, "I don't know." This is the case if we see either "blue, red" or "red, blue" for the hat colors of players 1 and 2. If we do see the colors of the other players matching the colors of a codeword (that is either "red, red" or "blue, blue"), then we have to take action by announcing the color that will direct us away from the codeword. If we see "blue, blue," we announce color "red." If we see "red, red," we announce color "blue."

The 16 Hamming codewords of a 1-error correcting perfect Hamming code with $n = 7$ players are shown in the figure on the following page. Appendix C.1 shows how to construct the codewords. But it is not too difficult to convince oneself that they indeed form a Hamming code. Each pair of codewords has at least the Hamming distance 3, which means that the code is 1-error correcting. It is a perfect code, as can be shown by a counting argument: Each ball of radius 1 around a codeword contains 8 sequences; in total, the balls contain $16 \cdot 8 = 128$ sequences. Therefore, any of the $2^7 = 128$ sequences of length 7 is contained in one of the balls.



This table can be used analogously to the shorter table shown in the previous figure. Let us say that we are player 3. If we observe for players 1, 2, 4, 5, 6, and 7 the colors "red, red, blue, red, red, and red" (in this order), then the colors match the codeword 9 (with the exception of our own color, which we cannot see). Again, in this scenario, we have to declare in this case the color that will steer us away from the codeword. The matching codeword 9 indicates color "red" for player 3, and therefore we have to declare that, in this case, we have a blue hat. If the colors of the other players do not match any of the 16 codewords, then we answer "I don't know."