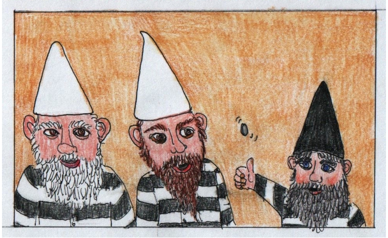
The Hat Problem

Brody Dylan Johnson
Saint Louis University



The Hat Problem[1]

A group of prisoners is allowed to play a game for their freedom. The prisoners are donned with either a black hat or a white hat and, while they cannot see their own hat, they can see the remaining hats. The two colors are equally likely. The prisoners play as a team and win when at least one prisoner guesses the color of his or her own hat without any incorrect guesses being made. The prisoners may strategize before the game, but cannot communicate in any fashion once the game begins.

[1] Todd Ebert, University of California, Santa Barbara (1998).

Important Aspects of the Game

Things to keep in mind:

Important Aspects of the Game

Things to keep in mind:

- Hat colors are independent events.

$$P(\text{White}) = P(\text{Black}) = \frac{1}{2}$$

Identify $\text{Black} = 0$ and $\text{White} = 1$.

Important Aspects of the Game

Things to keep in mind:

- Hat colors are independent events.

$$P(\textsc{White}) = P(\textsc{Black}) = \frac{1}{2}$$

  Identify $\textsc{Black} = 0$ and $\textsc{White} = 1$.

- An acceptable strategy must always result in at least one prisoner making a guess.

Important Aspects of the Game

Things to keep in mind:

- Hat colors are independent events.

$$P(\text{White}) = P(\text{Black}) = \frac{1}{2}$$

  Identify $\text{Black} = 0$ and $\text{White} = 1$.

- An acceptable strategy must always result in at least one prisoner making a guess.
- The prisoners WIN when at least one correct guess is made and no incorrect guesses are made.

Important Aspects of the Game

Things to keep in mind:

- Hat colors are independent events.

$$P(\textsc{White}) = P(\textsc{Black}) = \frac{1}{2}$$

  Identify $\textsc{Black} = 0$ and $\textsc{White} = 1$.
- An acceptable strategy must always result in at least one prisoner making a guess.
- The prisoners WIN when at least one correct guess is made and no incorrect guesses are made.
- A prisoner sees all of the hats, EXCEPT his or her own.

Important Aspects of the Game

Things to keep in mind:

- Hat colors are independent events.

$$P(\textsc{White}) = P(\textsc{Black}) = \frac{1}{2}$$

  Identify $\textsc{Black} = 0$ and $\textsc{White} = 1$.

- An acceptable strategy must always result in at least one prisoner making a guess.
- The prisoners WIN when at least one correct guess is made and no incorrect guesses are made.
- A prisoner sees all of the hats, EXCEPT his or her own.
- No communication is allowed between prisoners, including whether or not others have elected to GUESS or PASS.

Basic Observations

What strategy maximizes the chance of winning? What effect does the number of prisoners have on the game?

Basic Observations

What strategy maximizes the chance of winning? What effect does the number of prisoners have on the game?

- One prisoner: The prisoner is forced to guess and, thus, the probability of winning is simply $\frac{1}{2}$.

Basic Observations

What strategy maximizes the chance of winning? What effect does the number of prisoners have on the game?

- One prisoner: The prisoner is forced to guess and, thus, the probability of winning is simply $\frac{1}{2}$.
- Two prisoners: One can articulate a lot of strategies, but 50-50 is the best one can do.

Basic Observations

What strategy maximizes the chance of winning? What effect does the number of prisoners have on the game?

- One prisoner: The prisoner is forced to guess and, thus, the probability of winning is simply $\frac{1}{2}$.
- Two prisoners: One can articulate a lot of strategies, but 50-50 is the best one can do.
  - If both prisoners guess randomly their chance of winning is only $\frac{1}{4}$.

Basic Observations

What strategy maximizes the chance of winning? What effect does the number of prisoners have on the game?

- One prisoner: The prisoner is forced to guess and, thus, the probability of winning is simply $\frac{1}{2}$.
- Two prisoners: One can articulate a lot of strategies, but 50-50 is the best one can do.
  - If both prisoners guess randomly their chance of winning is only $\frac{1}{4}$.
  - A single prisoner guessing leads to a $\frac{1}{2}$ probability of victory.

Basic Observations

What strategy maximizes the chance of winning? What effect does the number of prisoners have on the game?

- One prisoner: The prisoner is forced to guess and, thus, the probability of winning is simply $\frac{1}{2}$.
- Two prisoners: One can articulate a lot of strategies, but 50-50 is the best one can do.
  - If both prisoners guess randomly their chance of winning is only $\frac{1}{4}$.
  - A single prisoner guessing leads to a $\frac{1}{2}$ probability of victory.

Is it clear that this is the optimal strategy for $n = 2$?

Three Prisoners

Three Prisoners

- Strategy: If a prisoner sees two hats of the same color (s)he guesses the opposite color. (Valid strategy because there must be two hats of the same color.)

Three Prisoners

- Strategy: If a prisoner sees two hats of the same color (s)he guesses the opposite color. (Valid strategy because there must be two hats of the same color.)
- Each possible outcome can be expressed as a 3-digit binary number:

$$\underbrace{\begin{matrix} 010 & 100 & 101 \\ 001 & 110 & 011 \end{matrix}}_{\text{WIN}} \quad \underbrace{\begin{matrix} 111 \\ 000 \end{matrix}}_{\text{LOSE}}$$

Thus, the probability of winning with this strategy is $\frac{3}{4}$.

Three Prisoners - Optimality

Proposition

The strategy just described for $n = 3$ is optimal.

Three Prisoners - Optimality

Proposition

The strategy just described for $n = 3$ is optimal.

Proof.

1. Each guess has a 50-50 chance of being correct.

Three Prisoners - Optimality

Proposition

The strategy just described for $n = 3$ is optimal.

Proof.

1. Each guess has a 50-50 chance of being correct.
2. Thus, among all possible outcomes there must be an equal number of correct and incorrect guesses.

Three Prisoners - Optimality

Proposition

The strategy just described for $n = 3$ is optimal.

Proof.

1. Each guess has a 50-50 chance of being correct.
2. Thus, among all possible outcomes there must be an equal number of correct and incorrect guesses.
3. Each WIN results from a single correct guess, while each LOSS stems from *three* incorrect guesses.

Three Prisoners - Optimality

Proposition

The strategy just described for $n = 3$ is optimal.

Proof.

1. Each guess has a 50-50 chance of being correct.
2. Thus, among all possible outcomes there must be an equal number of correct and incorrect guesses.
3. Each WIN results from a single correct guess, while each LOSS stems from *three* incorrect guesses.
4. Winning one more game requires one more correct guess (a total of 7), but now we have only 1 possible game to lose and 7 incorrect guesses to make. There are only 3 prisoners so this is impossible.

Upper Bound for $P(\textsc{Win})$

QUESTION: Can we find an upper limit on the probability of victory for N prisoners?

Upper Bound for $P(\textsc{Win})$

QUESTION: Can we find an upper limit on the probability of victory for N prisoners?

- Assume a single correct guess for each win and [up to] N incorrect guesses in each loss.

Upper Bound for $P(\textsc{Win})$

QUESTION: Can we find an upper limit on the probability of victory for N prisoners?

- Assume a single correct guess for each win and [up to] N incorrect guesses in each loss.
- It is possible to have X wins among the 2^N possible games provided that the losses allow room for X incorrect guesses, i.e.,

$$2^N - X \geq \frac{X}{N} \quad \text{or} \quad X \leq \frac{N \cdot 2^N}{N+1}.$$

Upper Bound for $P(\text{Win})$

QUESTION: Can we find an upper limit on the probability of victory for N prisoners?

- Assume a single correct guess for each win and [up to] N incorrect guesses in each loss.
- It is possible to have X wins among the 2^N possible games provided that the losses allow room for X incorrect guesses, i.e.,

$$2^N - X \geq \frac{X}{N} \quad \text{or} \quad X \leq \frac{N \cdot 2^N}{N+1}.$$

- This argument gives an upper bound on the probability of victory:

$$P(\text{Win}) \leq \frac{N}{N+1}.$$

Four Prisoners

Proposition

The optimal strategy for four prisoners is to reuse the three-player strategy, completely ignoring one of the prisoners.

Four Prisoners

Proposition

The optimal strategy for four prisoners is to reuse the three-player strategy, completely ignoring one of the prisoners.

Proof.

□

Four Prisoners

Proposition

The optimal strategy for four prisoners is to reuse the three-player strategy, completely ignoring one of the prisoners.

Proof.

1. The three-player strategy produces a win in 12 of the 16 possible outcomes, with 12 correct guesses in 12 wins and 12 incorrect guesses spread over the 4 losses.

□

Four Prisoners

Proposition

The optimal strategy for four prisoners is to reuse the three-player strategy, completely ignoring one of the prisoners.

Proof.

1. The three-player strategy produces a win in 12 of the 16 possible outcomes, with 12 correct guesses in 12 wins and 12 incorrect guesses spread over the 4 losses.
2. If a strategy leads to 13 wins in these 16 outcomes, then at least 13 incorrect guesses must be made in the 3 losses.

□

Four Prisoners

Proposition

The optimal strategy for four prisoners is to reuse the three-player strategy, completely ignoring one of the prisoners.

Proof.

1. The three-player strategy produces a win in 12 of the 16 possible outcomes, with 12 correct guesses in 12 wins and 12 incorrect guesses spread over the 4 losses.
2. If a strategy leads to 13 wins in these 16 outcomes, then at least 13 incorrect guesses must be made in the 3 losses.
3. Four prisoners can make only 12 guesses in 3 games, leading to a contradiction.

□

Five Prisoners: A Simple Approach

- With five prisoners, at least three hats must have the same color. (Moreover, at least two prisoners will see three hats of the same color.)

Five Prisoners: A Simple Approach

- With five prisoners, at least three hats must have the same color. (Moreover, at least two prisoners will see three hats of the same color.)
- The obvious generalization of the $n = 3$ strategy would be: if a prisoner sees three hats of the same color, (s)he guesses the opposite color.

Five Prisoners: A Simple Approach

- With five prisoners, at least three hats must have the same color. (Moreover, at least two prisoners will see three hats of the same color.)
- The obvious generalization of the $n = 3$ strategy would be: if a prisoner sees three hats of the same color, (s)he guesses the opposite color.
- How do we know when the team wins?

Five Prisoners: Too Simple?

- Assume that the first three hats are all white. Each of the other two prisoners will guess that their hat is black, so the only time they win is when both of those hats ARE black. In other words, they win when exactly three hats have the same color.

Five Prisoners: Too Simple?

- Assume that the first three hats are all white. Each of the other two prisoners will guess that their hat is black, so the only time they win is when both of those hats ARE black. In other words, they win when exactly three hats have the same color.
- There are $\binom{5}{3} = 10$ outcomes where three hats are white and an equal number where three hats are black. There are $2^5 = 32$ possible outcomes so the probability of winning with this strategy is $20/32 = \frac{5}{8}$.

Five Prisoners: Too Simple?

- Assume that the first three hats are all white. Each of the other two prisoners will guess that their hat is black, so the only time they win is when both of those hats ARE black. In other words, they win when exactly three hats have the same color.
- There are $\binom{5}{3} = 10$ outcomes where three hats are white and an equal number where three hats are black. There are $2^5 = 32$ possible outcomes so the probability of winning with this strategy is $20/32 = \frac{5}{8}$.
- This is worse than the $n = 3$ strategy $\frac{3}{4}$ and way worse than the upper bound of $\frac{5}{6}$. This is because the right/wrong guesses are not distributed effectively.

What Do We Know So Far?

- The following table summarizes what we know for $1 \leq N \leq 7$:

N	1	2	3	4	5	6	7
$P(\text{WIN})$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{3}{4}$	?	?	?
$N/(N+1)$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{4}{5}$	$\frac{5}{6}$	$\frac{6}{7}$	$\frac{7}{8}$

What Do We Know So Far?

- The following table summarizes what we know for $1 \leq N \leq 7$:

N	1	2	3	4	5	6	7
$P(\text{Win})$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{3}{4}$	?	?	?
$N/(N+1)$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{4}{5}$	$\frac{5}{6}$	$\frac{6}{7}$	$\frac{7}{8}$

- The theoretical maximum is achieved for $N = 1, 3$, but for $N = 2, 4$ there is no way to achieve the theoretical maximum.

What Do We Know So Far?

- The following table summarizes what we know for $1 \leq N \leq 7$:

N	1	2	3	4	5	6	7
$P(\text{WIN})$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{3}{4}$	?	?	?
$N/(N+1)$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{4}{5}$	$\frac{5}{6}$	$\frac{6}{7}$	$\frac{7}{8}$

- The theoretical maximum is achieved for $N = 1, 3$, but for $N = 2, 4$ there is no way to achieve the theoretical maximum.
- It is natural to wonder, then, for which N can one achieve the theoretical maximum? If not, what is the best one can do?

Vector Spaces over a Finite Field

We now develop some tools from linear algebra to better understand the three-player solution. The hope is that this will facilitate solutions for larger numbers of prisoners.

Vector Spaces over a Finite Field

We now develop some tools from linear algebra to better understand the three-player solution. The hope is that this will facilitate solutions for larger numbers of prisoners.

- The possible outcomes can be expressed using sequences of 1's and 0's. These sequences can be thought of finite-dimensional vectors over the field $\mathbb{F}_2 = \{0, 1\}$.

Vector Spaces over a Finite Field

We now develop some tools from linear algebra to better understand the three-player solution. The hope is that this will facilitate solutions for larger numbers of prisoners.

- The possible outcomes can be expressed using sequences of 1's and 0's. These sequences can be thought of finite-dimensional vectors over the field $\mathbb{F}_2 = \{0, 1\}$.
- It turns out that one can obtain strategies for the N-prisoner Hat Problem by examining subspaces of N-dimensional vector spaces over $\mathbb{F}_2$.

Vector Spaces over a Finite Field

We now develop some tools from linear algebra to better understand the three-player solution. The hope is that this will facilitate solutions for larger numbers of prisoners.

- The possible outcomes can be expressed using sequences of 1's and 0's. These sequences can be thought of finite-dimensional vectors over the field $\mathbb{F}_2 = \{0, 1\}$.
- It turns out that one can obtain strategies for the N-prisoner Hat Problem by examining subspaces of N-dimensional vector spaces over $\mathbb{F}_2$.
- Recall that the term *subspace* refers to a collection of vectors which is closed under addition and scalar multiplication.

Vector Spaces over $\mathbb{F}_2$

- Consider the field $\mathbb{F}_2 = \{0, 1\}$ with the addition/multiplication tables:

$+$	0	1
0	0	1
1	1	0

$\times$	0	1
0	0	0
1	0	1

Vector Spaces over $\mathbb{F}_2$

- Consider the field $\mathbb{F}_2 = \{0, 1\}$ with the addition/multiplication tables:

$+$	0	1
0	0	1
1	1	0

$\times$	0	1
0	0	0
1	0	1

- $\mathbb{F}_2^n$ is an n-dimensional vector space over $\mathbb{F}_2$. We may write vectors in $\mathbb{F}_2^n$ as n-digit binary numbers, but must be careful to add digit-wise. Example:

$$101 + 001 = 100 \qquad 101 + 001 \neq 110$$

Algebraic Coding Theory

- A *linear code*, $\mathcal{C}$, is a subspace of $\mathbb{F}_2^n$.

Algebraic Coding Theory

- A *linear code*, $\mathcal{C}$, is a subspace of $\mathbb{F}_2^n$.
- Such codes are used for data storage and transmission, e.g., on compact discs or in cellular communications.

Algebraic Coding Theory

- A *linear code*, $\mathcal{C}$, is a subspace of $\mathbb{F}_2^n$.
- Such codes are used for data storage and transmission, e.g., on compact discs or in cellular communications.
  - Data is encoded using vectors in the subspace comprising the code.

Algebraic Coding Theory

- A *linear code*, $\mathcal{C}$, is a subspace of $\mathbb{F}_2^n$.
- Such codes are used for data storage and transmission, e.g., on compact discs or in cellular communications.
  - Data is encoded using vectors in the subspace comprising the code.
  - Errors in reading/receiving the data result in vectors that are not in the code.

Algebraic Coding Theory

- A *linear code*, $\mathcal{C}$, is a subspace of $\mathbb{F}_2^n$.
- Such codes are used for data storage and transmission, e.g., on compact discs or in cellular communications.
  - Data is encoded using vectors in the subspace comprising the code.
  - Errors in reading/receiving the data result in vectors that are not in the code.
  - If the errors are small, there will only be one code vector which is *close* to the given vector, allowing for recovery of the data.

Algebraic Coding Theory:

- The *Hamming weight* of a vector $\vec{c} \in \mathbb{F}_2^n$, $w(\vec{c})$, is the number of non-zero digits in the vector.

Algebraic Coding Theory:

- The *Hamming weight* of a vector $\vec{c} \in \mathbb{F}_2^n$, $w(\vec{c})$, is the number of non-zero digits in the vector.
- The *Hamming distance* between $\vec{c}_1, \vec{c}_2 \in \mathbb{F}_2^n$ is

$$d(\vec{c}_1, \vec{c}_2) = w(\vec{c}_1 - \vec{c}_2).$$

This distance is the number of bits that must be *flipped* to switch one vector into the other.

Algebraic Coding Theory:

- The *Hamming weight* of a vector $\vec{c} \in \mathbb{F}_2^n$, $w(\vec{c})$, is the number of non-zero digits in the vector.
- The *Hamming distance* between $\vec{c}_1, \vec{c}_2 \in \mathbb{F}_2^n$ is

$$d(\vec{c}_1, \vec{c}_2) = w(\vec{c}_1 - \vec{c}_2).$$

  This distance is the number of bits that must be *flipped* to switch one vector into the other.
- Triangle Inequality: $d(\vec{c}_1, \vec{c}_3) \leq d(\vec{c}_1, \vec{c}_2) + d(\vec{c}_2, \vec{c}_3)$.

Algebraic Coding Theory:

- The *Hamming weight* of a vector $\vec{c} \in \mathbb{F}_2^n$, $w(\vec{c})$, is the number of non-zero digits in the vector.
- The *Hamming distance* between $\vec{c}_1, \vec{c}_2 \in \mathbb{F}_2^n$ is

$$d(\vec{c}_1, \vec{c}_2) = w(\vec{c}_1 - \vec{c}_2).$$

  This distance is the number of bits that must be *flipped* to switch one vector into the other.
- Triangle Inequality: $d(\vec{c}_1, \vec{c}_3) \leq d(\vec{c}_1, \vec{c}_2) + d(\vec{c}_2, \vec{c}_3)$.

Proof.

Each bit in which $\vec{c}_3$ differs from $\vec{c}_1$ falls into one of two groups: (a) $\vec{c}_3(k) = \vec{c}_2(k)$ which implies $\vec{c}_2(k) \neq \vec{c}_1(k)$ and (b) $\vec{c}_3(k) \neq \vec{c}_2(k)$. □

Three Prisoner Strategy:

- The code is $\mathcal{C} = \{000, 111\}$. The Hamming distance between the codewords is 3.

Three Prisoner Strategy:

- The code is $\mathcal{C} = \{000, 111\}$. The Hamming distance between the codewords is 3.
- Each non-codeword is Hamming distance one from a unique code word.

Three Prisoner Strategy:

- The code is $\mathcal{C} = \{000, 111\}$. The Hamming distance between the codewords is 3.
- Each non-codeword is Hamming distance one from a unique code word.
  - 000: 001, 010, 100.

Three Prisoner Strategy:

- The code is $\mathcal{C} = \{000, 111\}$. The Hamming distance between the codewords is 3.
- Each non-codeword is Hamming distance one from a unique code word.
  - 000: 001, 010, 100.
  - 111: 110, 101, 011.

Three Prisoner Strategy:

- The code is $\mathcal{C} = \{000, 111\}$. The Hamming distance between the codewords is 3.
- Each non-codeword is Hamming distance one from a unique code word.
  - 000: 001, 010, 100.
  - 111: 110, 101, 011.
- Each player is assigned a bit position.

Three Prisoner Strategy:

- The code is $\mathcal{C} = \{000, 111\}$. The Hamming distance between the codewords is 3.
- Each non-codeword is Hamming distance one from a unique code word.
  - 000: 001, 010, 100.
  - 111: 110, 101, 011.
- Each player is assigned a bit position.
  - If the two bits visible to the prisoner correspond to a codeword, the prisoner should guess the color to make a non-codeword.

Three Prisoner Strategy:

- The code is $\mathcal{C} = \{000, 111\}$. The Hamming distance between the codewords is 3.
- Each non-codeword is Hamming distance one from a unique code word.
  - 000: 001, 010, 100.
  - 111: 110, 101, 011.
- Each player is assigned a bit position.
  - If the two bits visible to the prisoner correspond to a codeword, the prisoner should guess the color to make a non-codeword.
  - If the two bits visible do not make a codeword, the prisoner does not guess.

(7,4) Hamming code:

- Let $n = 7$ and consider the code $\mathcal{C}$ corresponding to the subspace generated by $\{1000110, 0100101, 0010011, 0001111\}$. (Every codeword is a linear combination of these four vectors.)

(7,4) Hamming code:

- Let $n = 7$ and consider the code $\mathcal{C}$ corresponding to the subspace generated by $\{1000110, 0100101, 0010011, 0001111\}$. (Every codeword is a linear combination of these four vectors.)
- Our message will be the four coefficients used to construct a codeword as a linear combination. Let $\vec{a} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix}$. (4-digit binary message) We will transmit the codeword

$$\vec{c}_{\vec{a}} = \vec{a}G = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

(7,4) Hamming code:

- There is another subspace of $\mathbb{F}_2^7$, $\mathcal{C}^\perp$, so that the combined span of $\mathcal{C}$ and $\mathcal{C}^\perp$ is all of $\mathbb{F}_2^7$ and, moreover, every vector in $\mathcal{C}$ is orthogonal to every vector in $\mathcal{C}^\perp$. ($\mathcal{C}^\perp$ is 3-dimensional)

(7,4) Hamming code:

- There is another subspace of $\mathbb{F}_2^7$, $\mathcal{C}^\perp$, so that the combined span of $\mathcal{C}$ and $\mathcal{C}^\perp$ is all of $\mathbb{F}_2^7$ and, moreover, every vector in $\mathcal{C}$ is orthogonal to every vector in $\mathcal{C}^\perp$. ($\mathcal{C}^\perp$ is 3-dimensional)
- In our example $\mathcal{C}^\perp$ is generated by $\{1101100, 1011010, 0111001\}$. The orthogonality condition tells us that if $\vec{c} \in \mathcal{C}$, then

$$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix} = \vec{c}H^T = \vec{c}\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}^T.$$

(7,4) Hamming code:

- There is another subspace of $\mathbb{F}_2^7$, $\mathcal{C}^\perp$, so that the combined span of $\mathcal{C}$ and $\mathcal{C}^\perp$ is all of $\mathbb{F}_2^7$ and, moreover, every vector in $\mathcal{C}$ is orthogonal to every vector in $\mathcal{C}^\perp$. ($\mathcal{C}^\perp$ is 3-dimensional)
- In our example $\mathcal{C}^\perp$ is generated by $\{1101100, 1011010, 0111001\}$. The orthogonality condition tells us that if $\vec{c} \in \mathcal{C}$, then

$$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix} = \vec{c}H^T = \vec{c} \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}^T .$$

- Notice that columns 3, 6, and 7 of H sum to $\vec{0}$. Also observe that no *two* columns of H sum to $\vec{0}$.

How close can two codewords be?

Proposition

There is a nonzero codeword $\vec{c} \in \mathcal{C}$ such that $w(\vec{c}) \leq d$ if and only if there exists a set of d columns of H that are linearly dependent.

How close can two codewords be?

Proof.

□

How close can two codewords be?

Proof.

First observe that $\vec{c}H^T = \vec{0}$ is a fancy way of writing a linear combination of columns of H that equals $\vec{0}$.

How close can two codewords be?

Proof.

First observe that $\vec{c}H^T = \vec{0}$ is a fancy way of writing a linear combination of columns of H that equals $\vec{0}$.

($\Longrightarrow$): Suppose there is a vector $\vec{c}$ with $w(\vec{c}) \leq d$. The linear combination only involves $w(\vec{c}) \leq d$ columns and $\vec{c}H^T = \vec{0}$ is merely expressing the linear dependence of these $w(\vec{c})$ columns. □

How close can two codewords be?

Proof.

First observe that $\vec{c}H^T = \vec{0}$ is a fancy way of writing a linear combination of columns of H that equals $\vec{0}$.

($\Longrightarrow$): Suppose there is a vector $\vec{c}$ with $w(\vec{c}) \leq d$. The linear combination only involves $w(\vec{c}) \leq d$ columns and $\vec{c}H^T = \vec{0}$ is merely expressing the linear dependence of these $w(\vec{c})$ columns.

($\Longleftarrow$): Suppose there exist d columns $\vec{H}_{k_1}, \ldots, \vec{H}_{k_d}$ of H and scalars $a_{k_1}, \ldots, a_{k_d}$ (at least one nonzero) so that

$$\sum_{n=1}^{d} a_{k_n} \vec{H}_{k_n} = \vec{0}.$$

Choose $\vec{c}$ so that $\vec{c}(k_n) = a_{k_n}$ with $\vec{c}(k) = 0$ otherwise. Then $\vec{c}$ belongs to the code because $\vec{c}H^T = \vec{0}$ and $w(\vec{c}) \leq d$ by construction. □

$(7, 4)$ Hamming code:

- In our $(7, 4)$ Hamming code no two columns are linearly dependent, but there are triplets of columns which are dependent. Proposition 5.1 implies that no nonzero codeword $\vec{c}$ has $w(\vec{c}) \leq 2$.

$(7,4)$ Hamming code:

- In our $(7,4)$ Hamming code no two columns are linearly dependent, but there are triplets of columns which are dependent. Proposition 5.1 implies that no nonzero codeword $\vec{c}$ has $w(\vec{c}) \leq 2$.
- Recall that $d(\vec{c}_1, \vec{c}_2) = w(\vec{c}_1 - \vec{c}_2)$, so we know that $d(\vec{c}_1, \vec{c}_2) > 2$ for all the codewords in our code. In other words, we must change at least *three* bits to change one codeword into another.

$(7, 4)$ Hamming code:

- In our $(7, 4)$ Hamming code no two columns are linearly dependent, but there are triplets of columns which are dependent. Proposition 5.1 implies that no nonzero codeword $\vec{c}$ has $w(\vec{c}) \leq 2$.
- Recall that $d(\vec{c}_1, \vec{c}_2) = w(\vec{c}_1 - \vec{c}_2)$, so we know that $d(\vec{c}_1, \vec{c}_2) > 2$ for all the codewords in our code. In other words, we must change at least *three* bits to change one codeword into another.
- **Conclusion:** *If a word is one flip away from a codeword, then that codeword is the unique codeword closest to the given word.*

Dissecting the $(7, 4)$ Hamming code:

- How many codewords are there? 4-dimensional $\Rightarrow 2^4 = 16$ codewords

Dissecting the $(7, 4)$ Hamming code:

- How many codewords are there? 4-dimensional $\Rightarrow 2^4 = 16$ codewords
- How many words are one flip away from a given codeword? We have 7 digits, so there are seven words that differ from a codeword by 1. (None of these can be another codeword.)

Dissecting the $(7,4)$ Hamming code:

- How many codewords are there? 4-dimensional $\Rightarrow 2^4 = 16$ codewords
- How many words are one flip away from a given codeword? We have 7 digits, so there are seven words that differ from a codeword by 1. (None of these can be another codeword.)
- How many words do we have? $2^7 = 128 = 8 \times 16 = 16 + 7 \times 16.$

Dissecting the $(7,4)$ Hamming code:

- How many codewords are there? 4-dimensional $\Rightarrow 2^4 = 16$ codewords
- How many words are one flip away from a given codeword? We have 7 digits, so there are seven words that differ from a codeword by 1. (None of these can be another codeword.)
- How many words do we have? $2^7 = 128 = 8 \times 16 = 16 + 7 \times 16$.

Conclusion: *Every word is either a codeword or one flip away from a uniquely defined codeword.*

Seven Prisoner Strategy:

- Assign each guest a bit number and give each guest a list of codewords and the associated one-flip-away words. If a prisoner sees six correct digits then (s)he guesses that his or her bit will result in a non-codeword. If a prisoner sees that the word cannot be a codeword, (s)he simply elects not to guess.

Seven Prisoner Strategy:

- Assign each guest a bit number and give each guest a list of codewords and the associated one-flip-away words. If a prisoner sees six correct digits then (s)he guesses that his or her bit will result in a non-codeword. If a prisoner sees that the word cannot be a codeword, (s)he simply elects not to guess.
- LOSING: In 16 cases (the codewords) everyone sees 6 correct digits and guesses incorrectly that the full word is not a codeword. (All seven prisoners guess wrong.)

Seven Prisoner Strategy:

- Assign each guest a bit number and give each guest a list of codewords and the associated one-flip-away words. If a prisoner sees six correct digits then (s)he guesses that his or her bit will result in a non-codeword. If a prisoner sees that the word cannot be a codeword, (s)he simply elects not to guess.
- LOSING: In 16 cases (the codewords) everyone sees 6 correct digits and guesses incorrectly that the full word is not a codeword. (All seven prisoners guess wrong.)
- WINNING: In the remaining 112 cases only one digit actually is "wrong" so only the prisoner wearing this hat makes a guess. In this case the single guess is correct and the team wins.

Optimality and Extensions

- The probability of winning with this seven-player strategy is $\frac{112}{128} = \frac{7}{8}$. The optimality is evident because the theoretical upper bound is achieved.

Optimality and Extensions

- The probability of winning with this seven-player strategy is $\frac{112}{128} = \frac{7}{8}$. The optimality is evident because the theoretical upper bound is achieved.
- Hamming codes can be constructed for each dimension of the form $2^N - 1$, leading to optimal strategies for these numbers of players.

Optimality and Extensions

- The probability of winning with this seven-player strategy is $\frac{112}{128} = \frac{7}{8}$. The optimality is evident because the theoretical upper bound is achieved.
- Hamming codes can be constructed for each dimension of the form $2^N - 1$, leading to optimal strategies for these numbers of players.
- As N increases $P(\textsc{Win})$ approaches 1.

Conclusion

- A group of N prisoners can always employ a Hamming code strategy for a smaller number of the form $2^M - 1$.

Conclusion

- A group of N prisoners can always employ a Hamming code strategy for a smaller number of the form $2^M - 1$.
- Question: Are the Hamming code strategies optimal for all numbers of prisoners $N \geq 3$?

Conclusion

- A group of N prisoners can always employ a Hamming code strategy for a smaller number of the form $2^M - 1$.
- Question: Are the Hamming code strategies optimal for all numbers of prisoners $N \geq 3$?
  - One cannot improve upon the $N = 3$ strategy for four prisoners.

Conclusion

- A group of N prisoners can always employ a Hamming code strategy for a smaller number of the form $2^M - 1$.
- Question: Are the Hamming code strategies optimal for all numbers of prisoners $N \geq 3$?
  - One cannot improve upon the $N = 3$ strategy for four prisoners.
  - What about the five and six prisoner cases?

Conclusion

- A group of N prisoners can always employ a Hamming code strategy for a smaller number of the form $2^M - 1$.
- Question: Are the Hamming code strategies optimal for all numbers of prisoners $N \geq 3$?
  - One cannot improve upon the $N = 3$ strategy for four prisoners.
  - What about the five and six prisoner cases?
- If the hat colors are not equally likely, how will the optimal strategy be affected?