An Introduction to Coding Theory

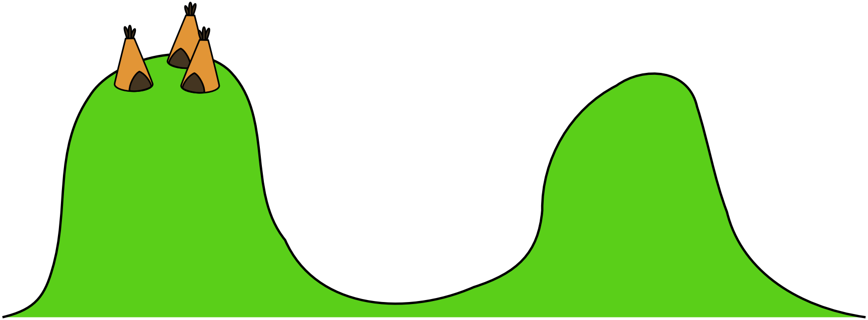
Nate Black

Clemson University

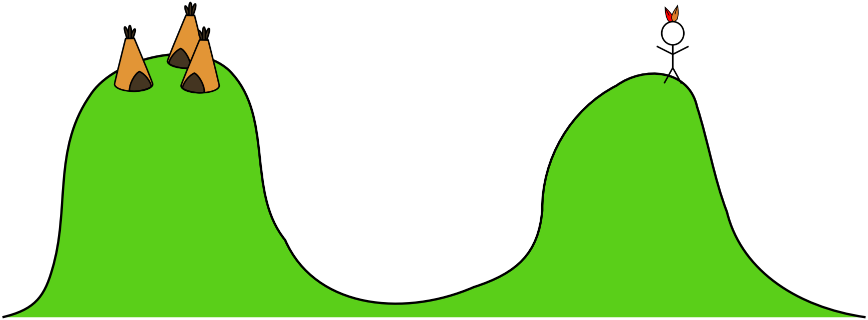
Graduate Student Seminar

November 2, 2011

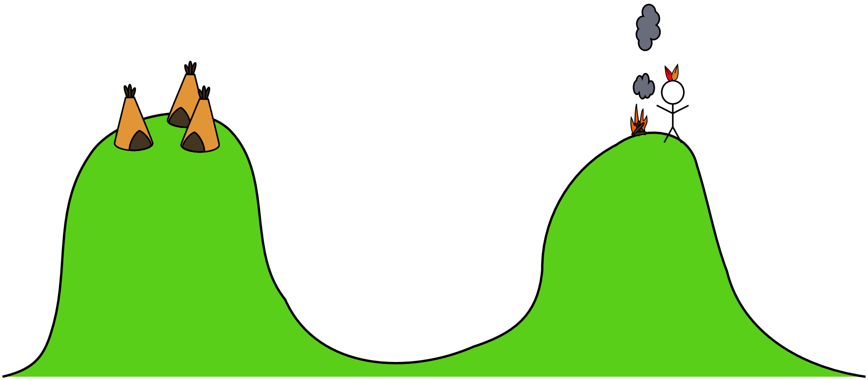
A small Indian village on one hill



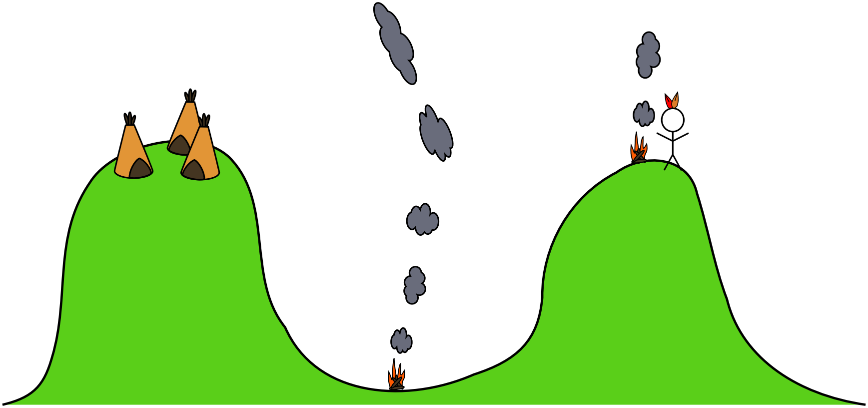
An Indian scout on another hill



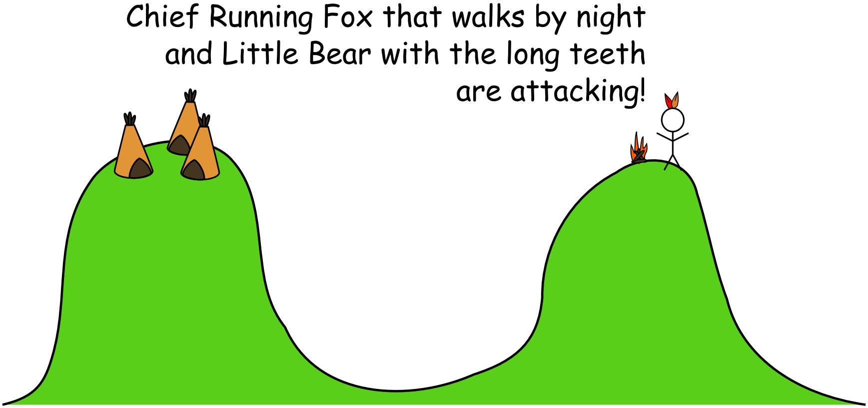
Communicating via smoke signals



Problem: noise



Problem: efficiency



Outline

- 1. Background Information
- 2. Error Correcting Codes
- 3. Efficiency via Compression
- 4. Applications

History

- 1948: *A Mathematical Theory of Communication* by Claude Shannon
- 1952: *A Method for the Construction of Minimum-Redundancy Codes* by David Huffman
- 1960: Reed-Solomon Codes introduced
- 1970: Goppa Codes introduced
- 1980s: Algebraic Geometry Codes popularized

History

- 1948: *A Mathematical Theory of Communication* by Claude Shannon
- 1952: *A Method for the Construction of Minimum-Redundancy Codes* by David Huffman
- 1960: Reed-Solomon Codes introduced
- 1970: Goppa Codes introduced
- 1980s: Algebraic Geometry Codes popularized

History

- 1948: *A Mathematical Theory of Communication* by Claude Shannon
- 1952: *A Method for the Construction of Minimum-Redundancy Codes* by David Huffman
- 1960: Reed-Solomon Codes introduced
- 1970: Goppa Codes introduced
- 1980s: Algebraic Geometry Codes popularized

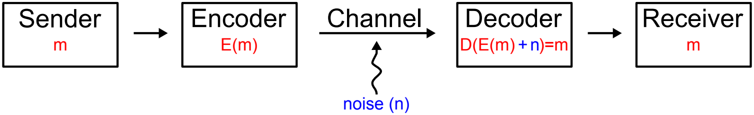
History

- 1948: *A Mathematical Theory of Communication* by Claude Shannon
- 1952: *A Method for the Construction of Minimum-Redundancy Codes* by David Huffman
- 1960: Reed-Solomon Codes introduced
- 1970: Goppa Codes introduced
- 1980s: Algebraic Geometry Codes popularized

History

- 1948: *A Mathematical Theory of Communication* by Claude Shannon
- 1952: *A Method for the Construction of Minimum-Redundancy Codes* by David Huffman
- 1960: Reed-Solomon Codes introduced
- 1970: Goppa Codes introduced
- 1980s: Algebraic Geometry Codes popularized

General Communication Model



General Definitions

- The **encoding alphabet**, A, is a set of symbols used to encode information.
- A **code**, C, is a subset of A^*, (i.e. all words over the alphabet A). The elements in this subset are called the **codewords** of C.
- A **block code** is a code where $C \subseteq A^n$.
- **Decoding** is the process of obtaining the original message from the received message.
- A **linear code**, is a block code where the subset C is a linear subspace of A^n. This additional structure makes decoding easier.

General Definitions

- The **encoding alphabet**, A, is a set of symbols used to encode information.
- A **code**, C, is a subset of A^*, (i.e. all words over the alphabet A). The elements in this subset are called the **codewords** of C.
- A **block code** is a code where $C \subseteq A^n$.
- **Decoding** is the process of obtaining the original message from the received message.
- A **linear code**, is a block code where the subset C is a linear subspace of A^n. This additional structure makes decoding easier.

General Definitions

- The **encoding alphabet**, A, is a set of symbols used to encode information.
- A **code**, C, is a subset of A^*, (i.e. all words over the alphabet A). The elements in this subset are called the **codewords** of C.
- A **block code** is a code where $C \subseteq A^n$.
- **Decoding** is the process of obtaining the original message from the received message.
- A **linear code**, is a block code where the subset C is a linear subspace of A^n. This additional structure makes decoding easier.

General Definitions

- The **encoding alphabet**, A, is a set of symbols used to encode information.
- A **code**, C, is a subset of A^*, (i.e. all words over the alphabet A). The elements in this subset are called the **codewords** of C.
- A **block code** is a code where $C \subseteq A^n$.
- **Decoding** is the process of obtaining the original message from the received message.
- A **linear code**, is a block code where the subset C is a linear subspace of A^n. This additional structure makes decoding easier.

General Definitions

- The **encoding alphabet**, A, is a set of symbols used to encode information.
- A **code**, C, is a subset of A^*, (i.e. all words over the alphabet A). The elements in this subset are called the **codewords** of C.
- A **block code** is a code where $C \subseteq A^n$.
- **Decoding** is the process of obtaining the original message from the received message.
- A **linear code**, is a block code where the subset C is a linear subspace of A^n. This additional structure makes decoding easier.

Toy Example

- Let $A = \{a, b, \ldots z\}$, and $C = \{aaa | a \in A\}$.
- Encode a message m by repeating each letter 3 times.
- Decode a received message by taking the most repeated character as the intended character. If they are all different, then output ?.
- Example:
  - The sender encodes *math* as a sequence of 4 words $\{mmm, aaa, ttt, hhh\}$.
  - The received messages are $\{msm, aaa, qtt, hrh\}$ due to some noise in the channel.
  - The decoder successfully recovers the message *math*.

Toy Example

- Let $A = \{a, b, \dots z\}$, and $C = \{aaa | a \in A\}$.
- Encode a message m by repeating each letter 3 times.
- Decode a received message by taking the most repeated character as the intended character. If they are all different, then output ?.
- Example:
  - The sender encodes *math* as a sequence of 4 words $\{mmm, aaa, ttt, hhh\}$.
  - The received messages are $\{msm, aaa, qtt, hrh\}$ due to some noise in the channel.
  - The decoder successfully recovers the message *math*.

Toy Example

- Let $A = \{a, b, \ldots z\}$, and $C = \{aaa | a \in A\}$.
- Encode a message m by repeating each letter 3 times.
- Decode a received message by taking the most repeated character as the intended character. If they are all different, then output ?.
- Example:
  - The sender encodes $math$ as a sequence of 4 words $\{mmm, aaa, ttt, hhh\}$.
  - The received messages are $\{msm, aaa, qtt, hrh\}$ due to some noise in the channel.
  - The decoder successfully recovers the message $math$.

Toy Example

- Let $A = \{a, b, \dots z\}$, and $C = \{aaa | a \in A\}$.
- Encode a message m by repeating each letter 3 times.
- Decode a received message by taking the most repeated character as the intended character. If they are all different, then output ?.
- Example:
  - The sender encodes *math* as a sequence of 4 words $\{mmm, aaa, ttt, hhh\}$.
  - The received messages are $\{msm, aaa, qtt, hrh\}$ due to some noise in the channel.
  - The decoder successfully recovers the message *math*.

Toy Example

- Let $A = \{a, b, \dots z\}$, and $C = \{aaa | a \in A\}$.
- Encode a message m by repeating each letter 3 times.
- Decode a received message by taking the most repeated character as the intended character. If they are all different, then output ?.
- Example:
  - The sender encodes *math* as a sequence of 4 words $\{mmm, aaa, ttt, hhh\}$.
  - The received messages are $\{msm, aaa, qtt, hrh\}$ due to some noise in the channel.
  - The decoder successfully recovers the message *math*.

Toy Example

- Let $A = \{a, b, \ldots z\}$, and $C = \{aaa | a \in A\}$.
- Encode a message m by repeating each letter 3 times.
- Decode a received message by taking the most repeated character as the intended character. If they are all different, then output ?.
- Example:
  - The sender encodes *math* as a sequence of 4 words $\{mmm, aaa, ttt, hhh\}$.
  - The received messages are $\{msm, aaa, qtt, hrh\}$ due to some noise in the channel.
  - The decoder successfully recovers the message *math*.

Toy Example

- Suppose the received messages were $\{msm, aaa, sts, hrq\}$.
- The decoder would return the message mas?.
- Decoding notes:
  - The decoder identified and corrected 1 error for m.
  - The decoder identified but couldn't correct the errors for h.
  - The decoder did not identify the errors for t and actually returned a wrong answer, s.

Toy Example

- Suppose the received messages were $\{msm, aaa, sts, hrq\}$.
- The decoder would return the message mas?.
- Decoding notes:
  - The decoder identified and corrected 1 error for m.
  - The decoder identified but couldn't correct the errors for h.
  - The decoder did not identify the errors for t and actually returned a wrong answer, s.

Toy Example

- Suppose the received messages were $\{msm, aaa, sts, hrq\}$.
- The decoder would return the message mas?.
- Decoding notes:
  - The decoder identified and corrected 1 error for m.
  - The decoder identified but couldn't correct the errors for h.
  - The decoder did not identify the errors for t and actually returned a wrong answer, s.

Toy Example

- Suppose the received messages were $\{msm, aaa, sts, hrq\}$.
- The decoder would return the message mas?.
- Decoding notes:
  - The decoder identified and corrected 1 error for m.
  - The decoder identified but couldn't correct the errors for h.
  - The decoder did not identify the errors for t and actually returned a wrong answer, s.

Toy Example

- Suppose the received messages were $\{msm, aaa, sts, hrq\}$.
- The decoder would return the message $mas?$.
- Decoding notes:
  - The decoder identified and corrected 1 error for m.
  - The decoder identified but couldn't correct the errors for h.
  - The decoder did not identify the errors for t and actually returned a wrong answer, s.

Error Correcting Codes

Linear Codes

- Normally we are interested in linear codes over finite fields such as $\mathbb{F}_2 = \{0, 1\}$.
- Notation: If a linear code C is a k-dimensional subspace of A^n with minimum distance d, then we say C is an $[n, k, d]$ linear code.

Linear Codes

- Normally we are interested in linear codes over finite fields such as $\mathbb{F}_2 = \{0, 1\}$.
- Notation: If a linear code C is a k-dimensional subspace of A^n with minimum distance d, then we say C is an $[n, k, d]$ linear code.

The Generator Matrix

Let $\{\mathbf{c}_1, \mathbf{c}_2, \dots \mathbf{c}_k\}$ be a basis for C, the k dimensional subspace of $\mathbb{F}^n$, where

$$\mathbf{c}_i = (c_{i,1},\ c_{i,2},\ \dots,\ c_{i,n})$$

is an n-vector. Then define the $k \times n$ matrix G as follows:

$$G = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_k \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{k,1} & c_{k,2} & \cdots & c_{k,n} \end{bmatrix}.$$

This matrix is called the generating matrix for C since $C = \left\{vG \mid v \in \mathbb{F}^k\right\}$ (i.e. all $\mathbb{F}$-linear combinations of the rows of G).

The Parity Check Matrix

Another related matrix which can be used to define the code C is the $(n-k) \times n$ matrix H of rank $n-k$ called the parity check matrix. This matrix is the solution to the following matrix equation:

$$GH^{\mathsf{T}} = 0_{k \times (n-k)},$$

and can be used in decoding received messages.

The Parity Check Matrix

Note that since every codeword, $\mathbf{v}$, can be written as $\mathbf{u}G = \mathbf{v}$ this implies that

$$\mathbf{v}H^{\mathsf{T}} = \mathbf{u}GH^{\mathsf{T}} = \mathbf{u}0_{k\times(n-k)} = 0_{1\times(n-k)}.$$

Also, since the rank of H is $n-k$ and $\{\mathbf{c}_1, \mathbf{c}_2, \ldots \mathbf{c}_k\} \subseteq C$ is a linearly independent set of size k with $\mathbf{c}_i H^{\mathsf{T}} = 0_{1\times(n-k)}$ we conclude that C is precisely the left null space of H^{T} and thus we have the following useful property:

$$\mathbf{v}H^{\mathsf{T}} = 0_{1\times(n-k)} \text{ iff } \mathbf{v} \in C.$$

Definition

- **Encoding**:
  Given $\mathbf{u} \in \mathbb{F}^k$ produce the corresponding codeword, $\mathbf{v} = \mathbf{u}G$.
- **Decoding**:
  Given $\mathbf{w} \in \mathbb{F}^n$ find the closest codeword, $\mathbf{c} \in C$.

Decoding Terms

Definition (Haming Weight)

Let $\mathbf{e} \in \mathbb{F}^n$, where $\mathbb{F}$ is a finite field. Then the hamming weight of $\mathbf{e}$, denoted $H(\mathbf{e})$ is the number of non-zero positions in $\mathbf{e}$.

Definition (Haming Distance)

Let $\mathbf{x}, \mathbf{y} \in \mathbb{F}^n$, where $\mathbb{F}$ is a finite field. Then the hamming distance between $\mathbf{x}$ and $\mathbf{y}$, denoted $H(\mathbf{x}, \mathbf{y})$ is the number of positions where $\mathbf{x}$ and $\mathbf{y}$ disagree. Thus, $H(\mathbf{x}, \mathbf{y}) = H(\mathbf{x} - \mathbf{y})$.

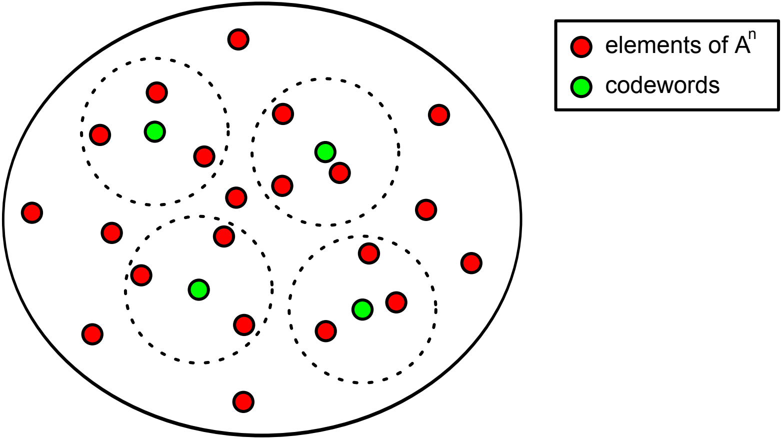
Examples

- If $\mathbf{x} = (0, 8, 1, 1, 2, 0, 0, 8)$, then $H(\mathbf{x}) = 5$.
- If $\mathbf{y} = (1, 1, 0, 2, 2, 0, 1, 1)$, then $H(\mathbf{x}, \mathbf{y}) = 6$.

Examples

- If $\mathbf{x} = (0, 8, 1, 1, 2, 0, 0, 8)$, then $H(\mathbf{x}) = 5$.
- If $\mathbf{y} = (1, 1, 0, 2, 2, 0, 1, 1)$, then $H(\mathbf{x}, \mathbf{y}) = 6$.

Decoding: Geometric Picture



Minimum Distance

Definition (Minimum Distance)

The minimum distance, d, of a code, C, is given by $d = \min(\{H(\mathbf{u}, \mathbf{v}) \mid \mathbf{u} \neq \mathbf{v} \text{ and } \mathbf{u}, \mathbf{v} \in C\})$.

- If C is a linear code then we have $H(\mathbf{u}, \mathbf{v}) = H(\mathbf{u} - \mathbf{v}, 0)$, since we can perform a distance preserving linear translation.
- Hence, for a linear code C, $d = min(\{H(\mathbf{w}, \mathbf{0}) \mid \mathbf{w} \in C\})$, and the minimum distance is the same as the minimum Hamming weight of all codewords.
- A code with minimum distance d can correct up to $\frac{d}{2}$ errors.

Minimum Distance

Definition (Minimum Distance)

The minimum distance, d, of a code, C, is given by
$d = \min\left(\{H(\mathbf{u},\mathbf{v}) \mid \mathbf{u} \neq \mathbf{v} \text{ and } \mathbf{u},\mathbf{v} \in C\}\right)$.

- If C is a linear code then we have $H(\mathbf{u},\mathbf{v}) = H(\mathbf{u}-\mathbf{v},0)$, since we can perform a distance preserving linear translation.
- Hence, for a linear code C, $d = min\left(\{H(\mathbf{w},\mathbf{0}) \mid \mathbf{w} \in C\}\right)$, and the minimum distance is the same as the minimum Hamming weight of all codewords.
- A code with minimum distance d can correct up to $\frac{d}{2}$ errors.

Minimum Distance

Definition (Minimum Distance)

The minimum distance, d, of a code, C, is given by
$d = \min(\{H(\mathbf{u}, \mathbf{v}) \mid \mathbf{u} \neq \mathbf{v} \text{ and } \mathbf{u}, \mathbf{v} \in C\})$.

- If C is a linear code then we have $H(\mathbf{u}, \mathbf{v}) = H(\mathbf{u} - \mathbf{v}, 0)$, since we can perform a distance preserving linear translation.
- Hence, for a linear code C, $d = min(\{H(\mathbf{w}, \mathbf{0}) \mid \mathbf{w} \in C\})$, and the minimum distance is the same as the minimum Hamming weight of all codewords.
- A code with minimum distance d can correct up to $\frac{d}{2}$ errors.

Minimum Distance

Definition (Minimum Distance)

The minimum distance, d, of a code, C, is given by
$d = \min\left(\{H(\mathbf{u}, \mathbf{v}) \mid \mathbf{u} \neq \mathbf{v} \text{ and } \mathbf{u}, \mathbf{v} \in C\}\right)$.

- If C is a linear code then we have $H(\mathbf{u}, \mathbf{v}) = H(\mathbf{u} - \mathbf{v}, 0)$, since we can perform a distance preserving linear translation.
- Hence, for a linear code C, $d = min\left(\{H(\mathbf{w}, \mathbf{0}) \mid \mathbf{w} \in C\}\right)$, and the minimum distance is the same as the minimum Hamming weight of all codewords.
- A code with minimum distance d can correct up to $\frac{d}{2}$ errors.

Singleton Bound

Theorem (Singleton Bound)

Let C be an $[n, k, d]$ linear code. Then $d \leq n - (k - 1)$.

Singleton Bound

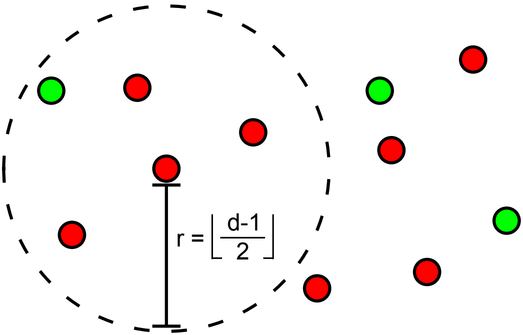
Theorem (Singleton Bound)

Let C be an $[n, k, d]$ linear code. Then $d \leq n - (k - 1)$.

Unambiguous Decoding

Definition (Unambiguous Decoding)

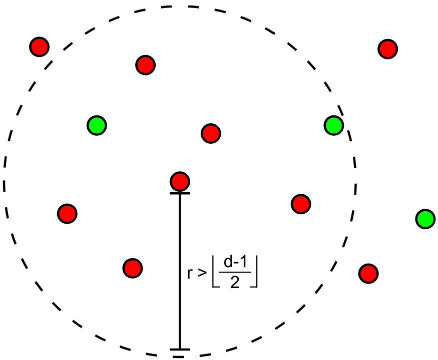
For an $[n, k, d]$ code and input $\mathbf{w} \in \mathbb{F}^n$, find the codeword, if it exists, within the ball of radius $r = \left\lfloor \frac{d-1}{2} \right\rfloor$ centered around $\mathbf{w}$.



List Decoding

Definition (List Decoding)

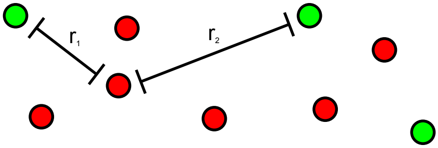
For an $[n, k, d]$ code and input $\mathbf{w} \in \mathbb{F}^n$, find all codewords, if any exist, within the ball of radius $r > \left\lfloor \frac{d-1}{2} \right\rfloor$ centered around $\mathbf{w}$.



Maximum Likelihood Decoding

Definition (Maximum Likelihood Decoding)

For an $[n, k, d]$ code and input $\mathbf{w} \in \mathbb{F}^n$, find the closest codeword to $\mathbf{w}$ with respect to the Hamming distance.



Decoding

Why Maximum Likelihood Decoding?

	vector components	distance
Received vector:	$[1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1]$	
Codeword 1:	$[1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1]$	3
Codeword 2:	$[1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0]$	4

Efficiency via Compression

Entropy

- Suppose **X** is a random variable taking on values from a finite set given by some fixed distribution.
- What is the most efficient way to encode the values that X takes on?
- Every language has a certain amount of redundancy built into it.

Definition (Entropy)

Let **X** be a random variable taking on values from a finite set X. The, **entropy** of **X** is given by

$$H(\mathbf{X}) = -\sum_{x \in X} \mathbf{Pr}[x] \log_2(\mathbf{Pr}[x])$$

Entropy

- Suppose **X** is a random variable taking on values from a finite set given by some fixed distribution.
- What is the most efficient way to encode the values that X takes on?
- Every language has a certain amount of redundancy built into it.

Definition (Entropy)

Let **X** be a random variable taking on values from a finite set X. The, **entropy** of **X** is given by

$$H(\mathbf{X}) = -\sum_{x \in X} \mathbf{Pr}[x] \log_2(\mathbf{Pr}[x])$$

Entropy

- Suppose **X** is a random variable taking on values from a finite set given by some fixed distribution.
- What is the most efficient way to encode the values that X takes on?
- Every language has a certain amount of redundancy built into it.

Definition (Entropy)

Let **X** be a random variable taking on values from a finite set X. The, **entropy** of **X** is given by

$$H(\mathbf{X}) = -\sum_{x \in X} \mathbf{Pr}[x] \log_2(\mathbf{Pr}[x])$$

Entropy

- Suppose $\mathbf{X}$ is a random variable taking on values from a finite set given by some fixed distribution.
- What is the most efficient way to encode the values that X takes on?
- Every language has a certain amount of redundancy built into it.

Definition (Entropy)

Let $\mathbf{X}$ be a random variable taking on values from a finite set X. The, **entropy** of $\mathbf{X}$ is given by

$$H(\mathbf{X}) = -\sum_{x \in X} \mathbf{Pr}[x] \log_2(\mathbf{Pr}[x])$$

Variable Length Codes

- We don't have to use the same size for each encoding of a message, as long as we can distinguish where one message ends and another begins.
- If we know that some messages occur more often, then we should use the fewest number of symbols possible to represent them.
- We should save the longest number of symbols for those messages that occur infrequently.

Variable Length Codes

- We don't have to use the same size for each encoding of a message, as long as we can distinguish where one message ends and another begins.
- If we know that some messages occur more often, then we should use the fewest number of symbols possible to represent them.
- We should save the longest number of symbols for those messages that occur infrequently.

Variable Length Codes

- We don't have to use the same size for each encoding of a message, as long as we can distinguish where one message ends and another begins.
- If we know that some messages occur more often, then we should use the fewest number of symbols possible to represent them.
- We should save the longest number of symbols for those messages that occur infrequently.

Huffman Encoding

- Suppose **X** takes on the values a, b, c with probability $\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$ respectively.
- We could encode them as follows:

$$\begin{aligned} a &\rightarrow 0 \\ b &\rightarrow 10 \\ c &\rightarrow 11 \end{aligned}$$

which would result in saving 1 character half of the time.

Huffman Encoding

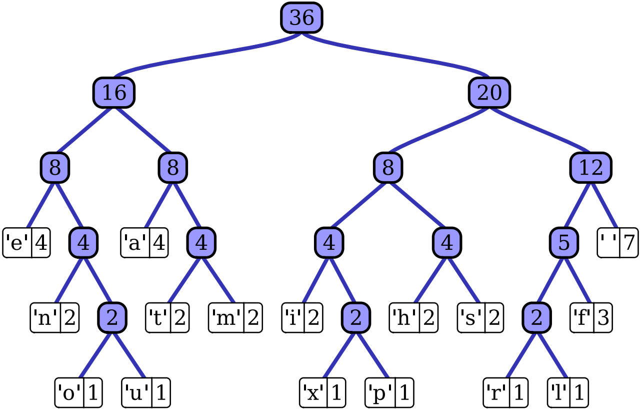
- Suppose **X** takes on the values a, b, c with probability $\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$ respectively.
- We could encode them as follows:

$$\begin{array}{ccc} a & \rightarrow & 0 \\ b & \rightarrow & 10 \\ c & \rightarrow & 11 \end{array}$$

which would result in saving 1 character half of the time.

Huffman Encoding

The symbols are stored in a frequency-sorted binary tree and the encoding is based off the "path" to the symbol.



Text: "this is an example of a huffman tree"

Applications

- Deep Space probe photos
- ISBN numbers and credit card numbers
- CDs
- Cryptography

Applications

- Deep Space probe photos
- ISBN numbers and credit card numbers
- CDs
- Cryptography

Applications

- Deep Space probe photos
- ISBN numbers and credit card numbers
- CDs
- Cryptography

Applications

- Deep Space probe photos
- ISBN numbers and credit card numbers
- CDs
- Cryptography

References

- Wikipedia (www.wikipedia.org)
- My Master's Project
- Various books from the library

Thanks for attending. If you are interested in this topic consider taking MTHSC 856 this spring.