NDMI011: Combinatorics and Graph Theory 1

Lecture #13

Error correcting codes

Irena Penev

January 4, 2020

- Suppose a sender wishes to send a message (say, a sequence of 1's and 0's) to a receiver.

- Suppose a sender wishes to send a message (say, a sequence of 1's and 0's) to a receiver.
- If the communication channel is unreliable or noisy, the message may get corrupted.
  - For instance, the sender may send 1011, and the receiver may get 1001.
  - The receiver does **not** get colored strings! We use red to emphasize errors.

- Suppose a sender wishes to send a message (say, a sequence of 1's and 0's) to a receiver.
- If the communication channel is unreliable or noisy, the message may get corrupted.
  - For instance, the sender may send 1011, and the receiver may get 1001.
  - The receiver does **not** get colored strings! We use red to emphasize errors.
- In this case, the receiver has no chance of spotting and fixing the error.

- Suppose a sender wishes to send a message (say, a sequence of 1's and 0's) to a receiver.
- If the communication channel is unreliable or noisy, the message may get corrupted.
  - For instance, the sender may send 1011, and the receiver may get 1001.
  - The receiver does **not** get colored strings! We use red to emphasize errors.
- In this case, the receiver has no chance of spotting and fixing the error.
- Can we help the receiver spot and fix errors?

- One strategy might be to agree to triple each bit (i.e. each 1 or 0). So, instead of 1011, we would send 111000111111.

- One strategy might be to agree to triple each bit (i.e. each 1 or 0). So, instead of 1011, we would send 111000111111.
- Suppose just one error occurred, and the receiver received 111000110111.

- One strategy might be to agree to triple each bit (i.e. each 1 or 0). So, instead of 1011, we would send 111000111111.
- Suppose just one error occurred, and the receiver received 111000110111.
- Now the receiver knows that there was an error in the boxed triple: 111000$\boxed{110}$111.

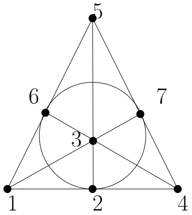
- One strategy might be to agree to triple each bit (i.e. each 1 or 0). So, instead of 1011, we would send 111000111111.
- Suppose just one error occurred, and the receiver received 111000110111.
- Now the receiver knows that there was an error in the boxed triple: 111000 $\boxed{110}$ 111.
- The receiver knows that the boxed triple should have been either 000 or 111, and the latter (i.e. 111) is more likely because it is more likely that only one error occurred than that two errors did.

- One strategy might be to agree to triple each bit (i.e. each 1 or 0). So, instead of 1011, we would send 111000111111.
- Suppose just one error occurred, and the receiver received 111000110111.
- Now the receiver knows that there was an error in the boxed triple: 111000 $\boxed{110}$ 111.
- The receiver knows that the boxed triple should have been either 000 or 111, and the latter (i.e. 111) is more likely because it is more likely that only one error occurred than that two errors did.
- So, the receiver guesses that the message sent was 111000111111, which corresponds to 1011.

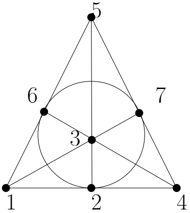
- One strategy might be to agree to triple each bit (i.e. each 1 or 0). So, instead of 1011, we would send 111000111111.
- Suppose just one error occurred, and the receiver received 111000110111.
- Now the receiver knows that there was an error in the boxed triple: 111000 $\boxed{110}$ 111.
- The receiver knows that the boxed triple should have been either 000 or 111, and the latter (i.e. 111) is more likely because it is more likely that only one error occurred than that two errors did.
- So, the receiver guesses that the message sent was 111000111111, which corresponds to 1011.
- However, if more than one error occurs in a triple corresponding to one bit, then the receiver will either fail to detect the error or will correct it incorrectly.

- One strategy might be to agree to triple each bit (i.e. each 1 or 0). So, instead of 1011, we would send 111000111111.
- Suppose just one error occurred, and the receiver received 111000110111.
- Now the receiver knows that there was an error in the boxed triple: 111000 $\boxed{110}$ 111.
- The receiver knows that the boxed triple should have been either 000 or 111, and the latter (i.e. 111) is more likely because it is more likely that only one error occurred than that two errors did.
- So, the receiver guesses that the message sent was 111000111111, which corresponds to 1011.
- However, if more than one error occurs in a triple corresponding to one bit, then the receiver will either fail to detect the error or will correct it incorrectly.
- For instance, if the receiver receives 111000100111, then he will incorrectly guess that the sender sent 111000000111, which corresponds to 1001.

- Here is another way to address the same problem. Consider the Fano plane (below).

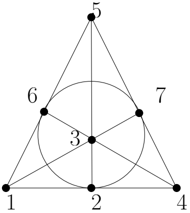


- Here is another way to address the same problem. Consider the Fano plane (below).



- We now form 16 row vectors of length seven as follows: we take all possible incidence vectors of lines of the Fano plane, the incidence vectors of the complements of the lines of the Fano plane, plus the vectors $(0, 0, 0, 0, 0, 0, 0)$ and $(1, 1, 1, 1, 1, 1, 1)$.
    - For example, the incidence vector of the line $\{1, 2, 4\}$ is $(1, 1, 0, 1, 0, 0, 0)$.
    - For example, the incidence vector of the complement of the line $\{1, 2, 4\}$ is $(0, 0, 1, 0, 1, 1, 1)$.

- Here is another way to address the same problem. Consider the Fano plane (below).



- We now form 16 row vectors of length seven as follows: we take all possible incidence vectors of lines of the Fano plane, the incidence vectors of the complements of the lines of the Fano plane, plus the vectors $(0, 0, 0, 0, 0, 0, 0)$ and $(1, 1, 1, 1, 1, 1, 1)$.
  - For example, the incidence vector of the line $\{1, 2, 4\}$ is $(1, 1, 0, 1, 0, 0, 0)$.
  - For example, the incidence vector of the complement of the line $\{1, 2, 4\}$ is $(0, 0, 1, 0, 1, 1, 1)$.
- Let $\mathcal{H}$ be the set of these 16 vectors.

- The 16 vectors in $\mathcal{H}$ have the following two properties:
  - any two distinct vectors in $\mathcal{H}$ differ in at least three places/coordinates;
  - for any vector $\mathbf{w}$ of 1's and 0's of length 7, there exists a unique vector $\mathbf{h} \in \mathcal{H}$ s.t. $\mathbf{w}$ and $\mathbf{h}$ differ in at most one place/coordinate.

- The 16 vectors in $\mathcal{H}$ have the following two properties:
  - any two distinct vectors in $\mathcal{H}$ differ in at least three places/coordinates;
  - for any vector $\mathbf{w}$ of 1's and 0's of length 7, there exists a unique vector $\mathbf{h} \in \mathcal{H}$ s.t. $\mathbf{w}$ and $\mathbf{h}$ differ in at most one place/coordinate.
- This means that if a sender sends a vector from $\mathcal{H}$, and at most one error is made during transmission, the receiver can correctly guess which vector was sent.
  - Indeed, the receiver simply chooses the unique vector from $\mathcal{H}$ that differs in at most one coordinate from the vector that the receiver received.

- The 16 vectors in $\mathcal{H}$ have the following two properties:
  - any two distinct vectors in $\mathcal{H}$ differ in at least three places/coordinates;
  - for any vector $\mathbf{w}$ of 1's and 0's of length 7, there exists a unique vector $\mathbf{h} \in \mathcal{H}$ s.t. $\mathbf{w}$ and $\mathbf{h}$ differ in at most one place/coordinate.
- This means that if a sender sends a vector from $\mathcal{H}$, and at most one error is made during transmission, the receiver can correctly guess which vector was sent.
  - Indeed, the receiver simply chooses the unique vector from $\mathcal{H}$ that differs in at most one coordinate from the vector that the receiver received.
- How do we use $\mathcal{H}$?

- There are precisely 16 stings of 1's and 0's of length four.

- There are precisely 16 stings of 1's and 0's of length four.
  - Indeed, these are simply the integers $0, 1, \ldots, 15$ written in binary code.
- So, we can set up a bijection π between the set of these 16 strings and the set $\mathcal{H}$.

- There are precisely 16 stings of 1's and 0's of length four.
  - Indeed, these are simply the integers $0, 1, \ldots, 15$ written in binary code.
- So, we can set up a bijection π between the set of these 16 strings and the set $\mathcal{H}$.
- Now, suppose we wish to transmit a string of 1's and 0's of length $4n$, for some positive integer n.

- There are precisely 16 stings of 1's and 0's of length four.
  - Indeed, these are simply the integers $0, 1, \dots, 15$ written in binary code.
- So, we can set up a bijection π between the set of these 16 strings and the set $\mathcal{H}$.
- Now, suppose we wish to transmit a string of 1's and 0's of length $4n$, for some positive integer n.
- We divide such a string into n consecutive blocks of length four, and instead of sending these blocks, we send (consecutively) the n vectors from $\mathcal{H}$ that correspond to them.

- There are precisely 16 stings of 1's and 0's of length four.
  - Indeed, these are simply the integers $0, 1, \ldots, 15$ written in binary code.
- So, we can set up a bijection π between the set of these 16 strings and the set $\mathcal{H}$.
- Now, suppose we wish to transmit a string of 1's and 0's of length $4n$, for some positive integer n.
- We divide such a string into n consecutive blocks of length four, and instead of sending these blocks, we send (consecutively) the n vectors from $\mathcal{H}$ that correspond to them.
- The advantage of this is that if, during transmission, at most one error is made in each vector, the receiver will be able to spot it and correct it, and then to read off (using π^{-1}) the sender's original $4n$-bit message.

- Note that, if we use $\mathcal{H}$, then instead of sending $4n$ bits (the number of bits in our original message), we send $7n$ bits.

- Note that, if we use $\mathcal{H}$, then instead of sending $4n$ bits (the number of bits in our original message), we send $7n$ bits.
- If data is expensive, then this is an improvement over tripling each bit (where we would send $3n$ bits for each n-bit message).

- Note that, if we use $\mathcal{H}$, then instead of sending $4n$ bits (the number of bits in our original message), we send $7n$ bits.
- If data is expensive, then this is an improvement over tripling each bit (where we would send $3n$ bits for each n-bit message).
- $\mathcal{H}$ is a type of "Hamming code," sometimes called the *Hamming(7,4) code* (because the original 4 bits are converted into 7 bits).

Definition

An *alphabet* is some finite set of symbols $\Sigma = \{s_0, \ldots, s_m\}$. A *word* of length n is a string (or row vector) of length n of symbols from our alphabet; Σ^n is the set of all words of length n using symbols from the alphabet Σ. A *code* is a subset C of Σ^n. Elements of the code are *codewords*.

Definition

An *alphabet* is some finite set of symbols $\Sigma = \{s_0, \ldots, s_m\}$. A *word* of length n is a string (or row vector) of length n of symbols from our alphabet; Σ^n is the set of all words of length n using symbols from the alphabet Σ. A *code* is a subset C of Σ^n. Elements of the code are *codewords*.

- Often, our alphabet is a finite field $\mathbb{F}_q$, where q is prime power.

Definition

An *alphabet* is some finite set of symbols $\Sigma = \{s_0, \ldots, s_m\}$. A *word* of length n is a string (or row vector) of length n of symbols from our alphabet; Σ^n is the set of all words of length n using symbols from the alphabet Σ. A *code* is a subset C of Σ^n. Elements of the code are *codewords*.

- Often, our alphabet is a finite field $\mathbb{F}_q$, where q is prime power.
  - Recall that, for a positive integer q, there is a field of size q iff q is a prime power (i.e. $q = p^n$, where p is a prime number and n is a positive integer).
  - All finite fields of the same size are isomorphic.
  - If q a prime power, then $\mathbb{F}_q$ is the unique (up to isomorphism) field of size q. Note that if p is a prime number, then $\mathbb{F}_p = \mathbb{Z}_p$ (but this is only true if p is prime!).

Definition

An *alphabet* is some finite set of symbols $\Sigma = \{s_0, \ldots, s_m\}$. A *word* of length n is a string (or row vector) of length n of symbols from our alphabet; Σ^n is the set of all words of length n using symbols from the alphabet Σ. A *code* is a subset C of Σ^n. Elements of the code are *codewords*.

- Often, our alphabet is a finite field $\mathbb{F}_q$, where q is prime power.
  - Recall that, for a positive integer q, there is a field of size q iff q is a prime power (i.e. $q = p^n$, where p is a prime number and n is a positive integer).
  - All finite fields of the same size are isomorphic.
  - If q a prime power, then $\mathbb{F}_q$ is the unique (up to isomorphism) field of size q. Note that if p is a prime number, then $\mathbb{F}_p = \mathbb{Z}_p$ (but this is only true if p is prime!).
- Particularly often, our alphabet is $\mathbb{F}_2 = \mathbb{Z}_2$, which is simply the binary code (and we can do addition and multiplication modulo 2).

Definition

Given words $\mathbf{x} = x_1 \dots x_n$ and $\mathbf{y} = y_1 \dots y_n$ in Σ^n (where Σ is some alphabet), the *Hamming distance* between $\mathbf{x}$ and $\mathbf{y}$, denoted by $d(\mathbf{x}, \mathbf{y})$, is the number of places in which $\mathbf{x}$ and $\mathbf{y}$ differ, i.e. $d(\mathbf{x}, \mathbf{y}) = |\{i \in \{1, \dots, n\} \mid x_i \neq y_i\}|$.

Definition

Given words $\mathbf{x} = x_1 \dots x_n$ and $\mathbf{y} = y_1 \dots y_n$ in Σ^n (where Σ is some alphabet), the *Hamming distance* between $\mathbf{x}$ and $\mathbf{y}$, denoted by $d(\mathbf{x}, \mathbf{y})$, is the number of places in which $\mathbf{x}$ and $\mathbf{y}$ differ, i.e. $d(\mathbf{x}, \mathbf{y}) = |\{i \in \{1, \dots, n\} \mid x_i \neq y_i\}|$.

- It is straightforward to check that the Hamming distance $d(\cdot, \cdot)$ is a "metric" on Σ^n, that is, that is satisfies the following three properties:
  - $d(x, y) = 0 \Leftrightarrow x = y$;
  - $d(x, y) = d(y, x)$;
  - $d(x, y) + d(y, z) \leq d(x, z)$.

  The inequality from the third bullet point is referred to as the *triangle inequality*.

- Codes are used as follows.
- A sender would like to send a message to a receiver, and for this, he uses some code $C \subseteq \Sigma^n$, where Σ is some alphabet.

- Codes are used as follows.
- A sender would like to send a message to a receiver, and for this, he uses some code $C \subseteq \Sigma^n$, where Σ is some alphabet.
- There is a bijection π (known both to the sender and the receiver) between all possible messages and the code C.

- Codes are used as follows.
- A sender would like to send a message to a receiver, and for this, he uses some code $C \subseteq \Sigma^n$, where Σ is some alphabet.
- There is a bijection π (known both to the sender and the receiver) between all possible messages and the code C.
- Now, the sender encodes his message (i.e. turns it into a codeword in a code via the bijection) and sends it to the receiver. The receiver receives this codeword, but possibly with some errors.
    - If the sender sends the codeword x and the receiver receives the word $\widetilde{x}$, then $d(x, \widetilde{x})$ is the number of errors created during transmission.

- Codes are used as follows.
- A sender would like to send a message to a receiver, and for this, he uses some code $C \subseteq \Sigma^n$, where Σ is some alphabet.
- There is a bijection π (known both to the sender and the receiver) between all possible messages and the code C.
- Now, the sender encodes his message (i.e. turns it into a codeword in a code via the bijection) and sends it to the receiver. The receiver receives this codeword, but possibly with some errors.
    - If the sender sends the codeword x and the receiver receives the word $\widetilde{x}$, then $d(x, \widetilde{x})$ is the number of errors created during transmission.
- The receiver corrects the errors (this is possible if the number of errors is small enough, where "small enough" depends on the code used), and then recovers the original message using π^{-1}.

- In general, there are two competing goals for codes.

- In general, there are two competing goals for codes.
- On the one hand, we wish to send as many different messages as possible, using as few bits as possible.

- In general, there are two competing goals for codes.
- On the one hand, we wish to send as many different messages as possible, using as few bits as possible.
- On the other hand, we wish to maximize the number of errors that we can successfully correct.

- Suppose Σ is an alphabet of size at least two, and $C \subseteq \Sigma^n$ is a code containing at least two codewords. Here are some parameters for the code C:
  - the codeword *length* is n;
  - the *size* of the alphabet is $q = |\Sigma|$;
  - the *dimension* of C is $|C|$, instead of which we often consider the logarithm $k = \log_q |C|$;
  - the *minimum distance* in C is $d = \min\{d(x, y) \mid x, y \in C, x \neq y\}$.

  A code with these parameters is an $(n, k, d)_q$*-code*.

- Suppose Σ is an alphabet of size at least two, and $C \subseteq \Sigma^n$ is a code containing at least two codewords. Here are some parameters for the code C:
  - the codeword *length* is n;
  - the *size* of the alphabet is $q = |\Sigma|$;
  - the *dimension* of C is $|C|$, instead of which we often consider the logarithm $k = \log_q |C|$;
  - the *minimum distance* in C is $d = \min\{d(x, y) \mid x, y \in C, x \neq y\}$.

  A code with these parameters is an $(n, k, d)_q$*-code*.
- Note that if at most $\lfloor \frac{d-1}{2} \rfloor$ errors are made during the transmission of a codeword, then the receiver can correctly spot and correct the errors by selecting the (unique) codeword with minimum Hamming distance from the word that he received.

- Suppose Σ is an alphabet of size at least two, and $C \subseteq \Sigma^n$ is a code containing at least two codewords. Here are some parameters for the code C:
  - the codeword *length* is n;
  - the *size* of the alphabet is $q = |\Sigma|$;
  - the *dimension* of C is $|C|$, instead of which we often consider the logarithm $k = \log_q |C|$;
  - the *minimum distance* in C is $d = \min\{d(x, y) \mid x, y \in C, x \neq y\}$.

  A code with these parameters is an $(n, k, d)_q$*-code*.

Example

The simplest code is the *total code* Σ^n, where Σ is an alphabet with $q = |\Sigma| \geq 2$ and n is a positive integer. The total code Σ^n is an $(n, n, 1)_q$ code. If we use this code, we send little data, but we cannot correct even a single error!

- Suppose Σ is an alphabet of size at least two, and $C \subseteq \Sigma^n$ is a code containing at least two codewords. Here are some parameters for the code C:
    - the codeword *length* is n;
    - the *size* of the alphabet is $q = |\Sigma|$;
    - the *dimension* of C is $|C|$, instead of which we often consider the logarithm $k = \log_q |C|$;
    - the *minimum distance* in C is $d = \min\{d(x, y) \mid x, y \in C, x \neq y\}$.

  A code with these parameters is an $(n, k, d)_q$*-code*.

Example

The *repetition code* Rep_n of length n over the alphabet Σ (with $q = |\Sigma| \geq 2$) is the code $C = \{\underbrace{x \dots x}_{n} \mid x \in \Sigma\}$. It is an $(n, 1, n)_q$-code. This code allows us to correct as many as $\lfloor \frac{n-1}{2} \rfloor$ errors, but it uses a lot of data.

- Suppose Σ is an alphabet of size at least two, and $C \subseteq \Sigma^n$ is a code containing at least two codewords. Here are some parameters for the code C:
  - the codeword *length* is n;
  - the *size* of the alphabet is $q = |\Sigma|$;
  - the *dimension* of C is $|C|$, instead of which we often consider the logarithm $k = \log_q |C|$;
  - the *minimum distance* in C is $d = \min\{d(x, y) \mid x, y \in C, x \neq y\}$.

  A code with these parameters is an $(n, k, d)_q$*-code*.

Example

The *parity code* C of length n (with $n \geq 2$) over the alphabet $\mathbb{F}_2$; it consists of all words of the form $w_1 \dots w_n$ with $w_1, \dots, w_n \in \mathbb{F}_2$ and $\sum_{i=1}^n w_i = 0$. It is an $(n, n-1, 2)_2$-code.[a]

[a] We have that $|C| = 2^{n-1}$, because the first $n-1$ symbols of a codeword can be chosen arbitrarily (and there are 2^{n-1} ways of doing this), but the n-th symbol is uniquely determined by the previous $n-1$ ones (because the sum must be 0). So, $k = \log_q |C| = \log_2 2^{n-1} = n-1$.

Definition

Given vectors $\mathbf{a} = (a_1, \ldots, a_n)^T$ and $\mathbf{b} = (b_1, \ldots, b_n)^T$ in $\mathbb{R}^n$, the *standard inner product* (or *dot product*) of $\mathbf{a}$ and $\mathbf{b}$ is $\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i$. Two vectors in $\mathbb{R}^n$ are *orthogonal* with respect to the dot product if their dot product is zero.

Definition

Given vectors $\mathbf{a} = (a_1, \ldots, a_n)^T$ and $\mathbf{b} = (b_1, \ldots, b_n)^T$ in $\mathbb{R}^n$, the *standard inner product* (or *dot product*) of $\mathbf{a}$ and $\mathbf{b}$ is $\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i$. Two vectors in $\mathbb{R}^n$ are *orthogonal* with respect to the dot product if their dot product is zero.

Definition

A *Hadamard matrix* of order n is an $n \times n$ matrix whose entries are all 1 or -1, and whose columns are pairwise orthogonal (with respect to the dot product).

Definition

A *Hadamard matrix* of order n is an $n \times n$ matrix whose entries are all 1 or -1, and whose columns are pairwise orthogonal (with respect to the dot product).

- For example, the matrix

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

  is Hadamard matrix of order 2.

- Furthermore, if H is an $n \times n$ Hadamard matrix, then

$$\begin{bmatrix} H & H \\ H & -H \end{bmatrix}$$

  is a Hadamard matrix of order $2n$.

Definition

A *Hadamard matrix* of order n is an $n \times n$ matrix whose entries are all 1 or -1, and whose columns are pairwise orthogonal (with respect to the dot product).

Proposition 2.1

Let H be a Hadamard matrix of order n. Then $HH^T = nI_n$.[a] Furthermore, H^T is also a Hadamard matrix of order n.

[a] As usual, I_n is the $n \times n$ identity matrix.

Proof. Lecture Notes.

Definition

If H is any Hadamard matrix of order n, then the Hadamard code associated with H consists of all rows of H and all rows of $-H$.

Definition

If H is any Hadamard matrix of order n, then the Hadamard code associated with H consists of all rows of H and all rows of $-H$.

- This code has $2n$ codewords.
  - For this, we must check that no two rows of H are the same, and that no row of H is equal to any row of $-H$. But this follows from the fact that, by Proposition 2.1, H^T is a Hadamard matrix (details?).

Definition

If H is any Hadamard matrix of order n, then the Hadamard code associated with H consists of all rows of H and all rows of $-H$.

- This code has $2n$ codewords.
  - For this, we must check that no two rows of H are the same, and that no row of H is equal to any row of $-H$. But this follows from the fact that, by Proposition 2.1, H^T is a Hadamard matrix (details?).
- It is easy to check that this is an $(n, 1 + \log_2 n, \frac{n}{2})_2$-code.

Definition

For positive integers n, d, q with $n \geq d$ and $q \geq 2$, let $A_q(n, d)$ be the maximum size of a code (i.e. the maximum possible number of codewords in a code) C with the following parameters:

- the size of the alphabet is q;
- the codeword length is n;
- the minimum distance is at least d.

The Singleton bound

For all positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, we have that $A_q(n, d) \leq q^{n-d+1}$.

Proof.

The Singleton bound

For all positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, we have that $A_q(n, d) \leq q^{n-d+1}$.

Proof. We prove this by induction on n, keeping q fixed and allowing d to vary.

The Singleton bound

For all positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, we have that $A_q(n, d) \leq q^{n-d+1}$.

Proof. We prove this by induction on n, keeping q fixed and allowing d to vary. More precisely, we fix positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, and we assume inductively that for all positive integers n', d' with $n' \geq d'$ and $n' < n$, we have that $A_q(n', d') \leq q^{n'-d'+1}$. We must show that $A_q(n, d) \leq q^{n-d+1}$.

The Singleton bound

For all positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, we have that $A_q(n,d) \leq q^{n-d+1}$.

Proof. We prove this by induction on n, keeping q fixed and allowing d to vary. More precisely, we fix positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, and we assume inductively that for all positive integers n', d' with $n' \geq d'$ and $n' < n$, we have that $A_q(n', d') \leq q^{n'-d'+1}$. We must show that $A_q(n,d) \leq q^{n-d+1}$.

Fix a code C over an alphabet Σ with $|\Sigma| = q$, and assume that the codeword length in C is n and that the minimum distance between codewords in C is at least d. We must show that $|C| \leq q^{n-d+1}$.

The Singleton bound

For all positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, we have that $A_q(n, d) \leq q^{n-d+1}$.

Proof. We prove this by induction on n, keeping q fixed and allowing d to vary. More precisely, we fix positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, and we assume inductively that for all positive integers n', d' with $n' \geq d'$ and $n' < n$, we have that $A_q(n', d') \leq q^{n'-d'+1}$. We must show that $A_q(n, d) \leq q^{n-d+1}$.

Fix a code C over an alphabet Σ with $|\Sigma| = q$, and assume that the codeword length in C is n and that the minimum distance between codewords in C is at least d. We must show that $|C| \leq q^{n-d+1}$. If $d = 1$, then

$$|C| \leq |\Sigma^n| = q^n = q^{n-d+1},$$

and we are done. So from now on, we assume that $d \geq 2$.

The Singleton bound

For all positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, we have that $A_q(n, d) \leq q^{n-d+1}$.

Proof (continued). Reminder: $C \subseteq \Sigma^n$, $|\Sigma| = q$, and the minimum distance between codewords in C is at least $d \geq 2$. WTS $|C| \leq q^{n-d+1}$.

The Singleton bound

For all positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, we have that $A_q(n, d) \leq q^{n-d+1}$.

Proof (continued). Reminder: $C \subseteq \Sigma^n$, $|\Sigma| = q$, and the minimum distance between codewords in C is at least $d \geq 2$. WTS $|C| \leq q^{n-d+1}$.

Let $\widetilde{C}$ be the set of all words $w_1 \dots w_{n-d+1}$ in Σ^{n-d+1} for which there exist some $w_{n-d+2}, \dots, w_n \in \Sigma$ s.t. $w_1 \dots w_{n-d+1} w_{n-d+2} \dots w_n \in C$.

The Singleton bound

For all positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, we have that $A_q(n, d) \leq q^{n-d+1}$.

Proof (continued). Reminder: $C \subseteq \Sigma^n$, $|\Sigma| = q$, and the minimum distance between codewords in C is at least $d \geq 2$. WTS $|C| \leq q^{n-d+1}$.

Let $\widetilde{C}$ be the set of all words $w_1 \dots w_{n-d+1}$ in Σ^{n-d+1} for which there exist some $w_{n-d+2}, \dots, w_n \in \Sigma$ s.t. $w_1 \dots w_{n-d+1} w_{n-d+2} \dots w_n \in C$. WTS $|\widetilde{C}| = |C|$.

Define $f : C \to \widetilde{C}$ by setting $f(w_1 \dots w_n) = w_1 \dots w_{n-d+1}$ for all $w_1 \dots w_n \in C$. WTS f is a bijection.

The Singleton bound

For all positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, we have that $A_q(n,d) \leq q^{n-d+1}$.

Proof (continued). Reminder: $C \subseteq \Sigma^n$, $|\Sigma| = q$, and the minimum distance between codewords in C is at least $d \geq 2$. WTS $|C| \leq q^{n-d+1}$.

Let $\widetilde{C}$ be the set of all words $w_1 \dots w_{n-d+1}$ in Σ^{n-d+1} for which there exist some $w_{n-d+2}, \dots, w_n \in \Sigma$ s.t. $w_1 \dots w_{n-d+1} w_{n-d+2} \dots w_n \in C$. WTS $|\widetilde{C}| = |C|$.

Define $f : C \to \widetilde{C}$ by setting $f(w_1 \dots w_n) = w_1 \dots w_{n-d+1}$ for all $w_1 \dots w_n \in C$. WTS f is a bijection. By construction, f is onto $\widetilde{C}$.

The Singleton bound

For all positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, we have that $A_q(n,d) \leq q^{n-d+1}$.

Proof (continued). Reminder: $C \subseteq \Sigma^n$, $|\Sigma| = q$, and the minimum distance between codewords in C is at least $d \geq 2$. WTS $|C| \leq q^{n-d+1}$.

Let $\widetilde{C}$ be the set of all words $w_1 \dots w_{n-d+1}$ in Σ^{n-d+1} for which there exist some $w_{n-d+2}, \dots, w_n \in \Sigma$ s.t. $w_1 \dots w_{n-d+1} w_{n-d+2} \dots w_n \in C$. WTS $|\widetilde{C}| = |C|$.

Define $f : C \to \widetilde{C}$ by setting $f(w_1 \dots w_n) = w_1 \dots w_{n-d+1}$ for all $w_1 \dots w_n \in C$. WTS f is a bijection. By construction, f is onto $\widetilde{C}$. Fix codewords $\mathbf{w} = w_1 \dots w_n$ and $\mathbf{w}' = w_1' \dots w_n'$ in C s.t. $f(\mathbf{w}) = f(\mathbf{w}')$; then $w_1 \dots w_{n-d+1} = w_1' \dots w_{n-d+1}'$, and so $d(\mathbf{w}, \mathbf{w}') \leq d - 1$.

The Singleton bound

For all positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, we have that $A_q(n, d) \leq q^{n-d+1}$.

Proof (continued). Reminder: $C \subseteq \Sigma^n$, $|\Sigma| = q$, and the minimum distance between codewords in C is at least $d \geq 2$. WTS $|C| \leq q^{n-d+1}$.

Let $\widetilde{C}$ be the set of all words $w_1 \dots w_{n-d+1}$ in Σ^{n-d+1} for which there exist some $w_{n-d+2}, \dots, w_n \in \Sigma$ s.t. $w_1 \dots w_{n-d+1} w_{n-d+2} \dots w_n \in C$. WTS $|\widetilde{C}| = |C|$.

Define $f : C \to \widetilde{C}$ by setting $f(w_1 \dots w_n) = w_1 \dots w_{n-d+1}$ for all $w_1 \dots w_n \in C$. WTS f is a bijection. By construction, f is onto $\widetilde{C}$. Fix codewords $\mathbf{w} = w_1 \dots w_n$ and $\mathbf{w}' = w'_1 \dots w'_n$ in C s.t. $f(\mathbf{w}) = f(\mathbf{w}')$; then $w_1 \dots w_{n-d+1} = w'_1 \dots w'_{n-d+1}$, and so $d(\mathbf{w}, \mathbf{w}') \leq d - 1$. Since the minimum distance in C is at least d, we conclude that $\mathbf{w} = \mathbf{w}'$. So, f is one-to-one.

The Singleton bound

For all positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, we have that $A_q(n, d) \leq q^{n-d+1}$.

Proof (continued). Reminder: $C \subseteq \Sigma^n$, $|\Sigma| = q$, and the minimum distance between codewords in C is at least $d \geq 2$. WTS $|C| \leq q^{n-d+1}$.

Let $\widetilde{C}$ be the set of all words $w_1 \dots w_{n-d+1}$ in Σ^{n-d+1} for which there exist some $w_{n-d+2}, \dots, w_n \in \Sigma$ s.t. $w_1 \dots w_{n-d+1} w_{n-d+2} \dots w_n \in C$. WTS $|\widetilde{C}| = |C|$.

Define $f : C \to \widetilde{C}$ by setting $f(w_1 \dots w_n) = w_1 \dots w_{n-d+1}$ for all $w_1 \dots w_n \in C$. WTS f is a bijection. By construction, f is onto $\widetilde{C}$. Fix codewords $\mathbf{w} = w_1 \dots w_n$ and $\mathbf{w}' = w'_1 \dots w'_n$ in C s.t. $f(\mathbf{w}) = f(\mathbf{w}')$; then $w_1 \dots w_{n-d+1} = w'_1 \dots w'_{n-d+1}$, and so $d(\mathbf{w}, \mathbf{w}') \leq d - 1$. Since the minimum distance in C is at least d, we conclude that $\mathbf{w} = \mathbf{w}'$. So, f is one-to-one. Thus, $f : C \to \widetilde{C}$ is a bijection, and we deduce that $|\widetilde{C}| = |C|$.

The Singleton bound

For all positive integers n, d, q s.t. $n \geq d$ and $q \geq 2$, we have that $A_q(n, d) \leq q^{n-d+1}$.

Proof (continued). Now, $\widetilde{C}$ is a code over Σ, with $|\Sigma| = q$, the length of codewords in $\widetilde{C}$ is $n - d + 1 < n$, and obviously, the minimum distance in $\widetilde{C}$ is at least 1. So, by the induction hypothesis, we have that

$$|\widetilde{C}| \leq A_q(n - d + 1, 1) \leq q^{(n-d+1)-1+1} = q^{n-d+1}.$$

Since $|\widetilde{C}| = |C|$, we deduce that $|C| \leq q^{n-d+1}$, which is what we needed to show.

Definition

Suppose n, t, q are positive integers and Σ is an alphabet of size q. For all $\mathbf{w} \in \Sigma^n$, we let $B_t^{\Sigma^n}(\mathbf{w})$ be the "combinatorial ball" of radius t around $\mathbf{w}$, i.e. $B_t^{\Sigma^n}(\mathbf{w})$ is the set of all words in Σ^n whose Hamming distance from $\mathbf{w}$ is at most t. When no confusion is possible, we write $B_t(\mathbf{w})$ instead of $B_t^{\Sigma^n}(\mathbf{w})$.

Definition

Suppose n, t, q are positive integers and Σ is an alphabet of size q. For all $\mathbf{w} \in \Sigma^n$, we let $B_t^{\Sigma^n}(\mathbf{w})$ be the "combinatorial ball" of radius t around $\mathbf{w}$, i.e. $B_t^{\Sigma^n}(\mathbf{w})$ is the set of all words in Σ^n whose Hamming distance from $\mathbf{w}$ is at most t. When no confusion is possible, we write $B_t(\mathbf{w})$ instead of $B_t^{\Sigma^n}(\mathbf{w})$.

Proposition 3.1

Let n, t, q be positive integers s.t. $n \geq t$ and $q \geq 2$, and let Σ be an alphabet of size q. Then $|B_t(\mathbf{w})| = \sum_{k=0}^{t} \binom{n}{k}(q-1)^k$ for all $\mathbf{w} \in \Sigma^n$.

Proposition 3.1

Let n, t, q be positive integers s.t. $n \geq t$ and $q \geq 2$, and let Σ be an alphabet of size q. Then $|B_t(\mathbf{w})| = \sum_{k=0}^{t} \binom{n}{k}(q-1)^k$ for all $\mathbf{w} \in \Sigma^n$.

Proof. Fix a word $\mathbf{w} \in \Sigma^n$. We must show that the number of words in Σ^n at distance at most t from $\mathbf{w}$ is precisely $\sum_{k=0}^{t} \binom{n}{k}(q-1)^k$. Clearly, it suffices to show that for all $k \in \{0, \dots, t\}$, the number of words in Σ^n at distance k from $\mathbf{w}$ is precisely $\binom{n}{k}(q-1)^k$. So, fix $k \in \{0, \dots, t\}$. There are $\binom{n}{k}$ ways to choose the k places in which a word at Hamming distance k from $\mathbf{w}$ differs from $\mathbf{w}$. For each such choice, and for each of the k selected placed, we have $q-1$ ways of altering $\mathbf{w}$ in that place;[1] so, for all k places together, we get $(q-1)^k$ ways of altering $\mathbf{w}$. So, there are precisely $\binom{n}{k}(q-1)^k$ words in Σ^n at distance k from $\mathbf{w}$.

[1] Indeed, we can select any symbol from Σ, except the one that appears in the selected place in the word $\mathbf{w}$ itself. Since $|\Sigma| = q$, we have $q-1$ choices.

The Hamming bound

Let n, d, q be positive integers such that $n \geq d$ and $q \geq 2$, and let $t = \lfloor \frac{d-1}{2} \rfloor$. Then $A_q(n, d) \leq \frac{q^n}{\sum_{k=0}^{t} \binom{n}{k}(q-1)^k}$.

Proof.

The Hamming bound

Let n, d, q be positive integers such that $n \geq d$ and $q \geq 2$, and let $t = \lfloor \frac{d-1}{2} \rfloor$. Then $A_q(n,d) \leq \frac{q^n}{\sum_{k=0}^{t} \binom{n}{k}(q-1)^k}$.

Proof. Fix a code $C \subseteq \Sigma^n$, where Σ is an alphabet of size q, and assume that the minimum distance between codewords in C is at least d. We must show that $|C| \leq \frac{q^n}{\sum_{k=0}^{t} \binom{n}{k}(q-1)^k}$.

The Hamming bound

Let n, d, q be positive integers such that $n \geq d$ and $q \geq 2$, and let $t = \lfloor \frac{d-1}{2} \rfloor$. Then $A_q(n,d) \leq \frac{q^n}{\sum_{k=0}^{t} \binom{n}{k}(q-1)^k}$.

Proof. Fix a code $C \subseteq \Sigma^n$, where Σ is an alphabet of size q, and assume that the minimum distance between codewords in C is at least d. We must show that $|C| \leq \frac{q^n}{\sum_{k=0}^{t} \binom{n}{k}(q-1)^k}$. Set $m = |C|$ and $C = \{\mathbf{c}_1, \dots, \mathbf{c}_m\}$. Since the minimum Hamming distance between codewords in C is at least d, and since $t = \lfloor \frac{d-1}{2} \rfloor$, we see that the combinatorial balls $B_t(\mathbf{c}_1), \dots, B_t(\mathbf{c}_m)$ are pairwise disjoint.

The Hamming bound

Let n, d, q be positive integers such that $n \geq d$ and $q \geq 2$, and let $t = \lfloor \frac{d-1}{2} \rfloor$. Then $A_q(n, d) \leq \frac{q^n}{\sum_{k=0}^{t} \binom{n}{k}(q-1)^k}$.

Proof. Fix a code $C \subseteq \Sigma^n$, where Σ is an alphabet of size q, and assume that the minimum distance between codewords in C is at least d. We must show that $|C| \leq \frac{q^n}{\sum_{k=0}^{t} \binom{n}{k}(q-1)^k}$. Set $m = |C|$ and $C = \{\mathbf{c}_1, \dots, \mathbf{c}_m\}$. Since the minimum Hamming distance between codewords in C is at least d, and since $t = \lfloor \frac{d-1}{2} \rfloor$, we see that the combinatorial balls $B_t(\mathbf{c}_1), \dots, B_t(\mathbf{c}_m)$ are pairwise disjoint. We now compute:

The Hamming bound

Let n, d, q be positive integers such that $n \geq d$ and $q \geq 2$, and let $t = \lfloor \frac{d-1}{2} \rfloor$. Then $A_q(n,d) \leq \frac{q^n}{\sum_{k=0}^{t} \binom{n}{k}(q-1)^k}$.

Proof (continued).

$$
\begin{aligned}
q^n &= |\Sigma^n| && \text{because } |\Sigma| = q \\
&\geq \left|\bigcup_{i=1}^{m} B_t(\mathbf{c}_i)\right| \\
&= \sum_{i=1}^{m} |B_t(\mathbf{c}_i)| && \text{because } B_t(\mathbf{c}_1), \dots, B_t(\mathbf{c}_m) \\
& && \text{are pairwise disjoint} \\
&= m \sum_{k=0}^{t} \binom{n}{k}(q-1)^k && \text{by Proposition 3.1} \\
&= |C| \sum_{k=0}^{t} \binom{n}{k}(q-1)^k && \text{because } m = |C|
\end{aligned}
$$

This implies that $|C| \leq \frac{q^n}{\sum_{k=0}^{t} \binom{n}{k}(q-1)^k}$, which is what we needed to show.

The Gilbert-Varshamov bound

Let n, d, q be positive integers such that $n \geq d$ and $q \geq 2$. Then $A_q(n, d) \geq \frac{q^n}{\sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k}$.

Proof.

The Gilbert-Varshamov bound

Let n, d, q be positive integers such that $n \geq d$ and $q \geq 2$. Then $A_q(n, d) \geq \frac{q^n}{\sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k}$.

Proof. Fix a code $C \subseteq \Sigma^n$, where Σ is some alphabet of size q, with minimum distance between codewords in C at least d, and with $|C| = A_q(n, d)$. WTS $|C| \geq \frac{q^n}{\sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k}$.

The Gilbert-Varshamov bound

Let n, d, q be positive integers such that $n \geq d$ and $q \geq 2$. Then $A_q(n,d) \geq \frac{q^n}{\sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k}$.

Proof. Fix a code $C \subseteq \Sigma^n$, where Σ is some alphabet of size q, with minimum distance between codewords in C at least d, and with $|C| = A_q(n,d)$. WTS $|C| \geq \frac{q^n}{\sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k}$.

Set $m = |C|$ and $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_m\}$.

Claim. $\Sigma^n = \bigcup_{i=1}^m B_{d-1}(\mathbf{c}_i)$.

The Gilbert-Varshamov bound

Let n, d, q be positive integers such that $n \geq d$ and $q \geq 2$. Then $A_q(n,d) \geq \frac{q^n}{\sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k}$.

Proof. Fix a code $C \subseteq \Sigma^n$, where Σ is some alphabet of size q, with minimum distance between codewords in C at least d, and with $|C| = A_q(n,d)$. WTS $|C| \geq \frac{q^n}{\sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k}$.

Set $m = |C|$ and $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_m\}$.

Claim. $\Sigma^n = \bigcup_{i=1}^m B_{d-1}(\mathbf{c}_i)$.

Proof of the Claim. If $\bigcup_{i=1}^m B_{d-1}(\mathbf{c}_i) \subsetneq \Sigma^n$, then fix some $\mathbf{w} \in \Sigma^n \setminus \left(\bigcup_{i=1}^m B_{d-1}(\mathbf{c}_i)\right)$. Then $d(\mathbf{w}, \mathbf{c}_i) \geq d$ for all $i \in \{1, \ldots, m\}$.

The Gilbert-Varshamov bound

Let n, d, q be positive integers such that $n \geq d$ and $q \geq 2$. Then $A_q(n,d) \geq \frac{q^n}{\sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k}$.

Proof. Fix a code $C \subseteq \Sigma^n$, where Σ is some alphabet of size q, with minimum distance between codewords in C at least d, and with $|C| = A_q(n,d)$. WTS $|C| \geq \frac{q^n}{\sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k}$.

Set $m = |C|$ and $C = \{\mathbf{c}_1, \dots, \mathbf{c}_m\}$.

Claim. *$\Sigma^n = \bigcup_{i=1}^m B_{d-1}(\mathbf{c}_i)$.*

Proof of the Claim. If $\bigcup_{i=1}^m B_{d-1}(\mathbf{c}_i) \subsetneqq \Sigma^n$, then fix some $\mathbf{w} \in \Sigma^n \setminus \left(\bigcup_{i=1}^m B_{d-1}(\mathbf{c}_i)\right)$. Then $d(\mathbf{w}, \mathbf{c}_i) \geq d$ for all $i \in \{1, \dots, m\}$. We now form a new code $\widetilde{C} := C \cup \{\mathbf{w}\}$; obviously, $\widetilde{C} \subseteq \Sigma^n$, with $|\Sigma| = q$, and by construction, the minimum distance in $\widetilde{C}$ is at least d.

The Gilbert-Varshamov bound

Let n, d, q be positive integers such that $n \geq d$ and $q \geq 2$. Then $A_q(n,d) \geq \frac{q^n}{\sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k}$.

Proof. Fix a code $C \subseteq \Sigma^n$, where Σ is some alphabet of size q, with minimum distance between codewords in C at least d, and with $|C| = A_q(n,d)$. WTS $|C| \geq \frac{q^n}{\sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k}$.

Set $m = |C|$ and $C = \{\mathbf{c}_1, \dots, \mathbf{c}_m\}$.

Claim. *$\Sigma^n = \bigcup_{i=1}^m B_{d-1}(\mathbf{c}_i)$.*

Proof of the Claim. If $\bigcup_{i=1}^m B_{d-1}(\mathbf{c}_i) \subsetneqq \Sigma^n$, then fix some $\mathbf{w} \in \Sigma^n \setminus \left(\bigcup_{i=1}^m B_{d-1}(\mathbf{c}_i) \right)$. Then $d(\mathbf{w}, \mathbf{c}_i) \geq d$ for all $i \in \{1, \dots, m\}$. We now form a new code $\widetilde{C} := C \cup \{\mathbf{w}\}$; obviously, $\widetilde{C} \subseteq \Sigma^n$, with $|\Sigma| = q$, and by construction, the minimum distance in $\widetilde{C}$ is at least d. But now the fact that $|\widetilde{C}| = |C| + 1 = A_q(n,d) + 1$ contradicts the definition of $A_q(n,d)$.

The Gilbert-Varshamov bound

Let n, d, q be positive integers such that $n \geq d$ and $q \geq 2$. Then $A_q(n,d) \geq \frac{q^n}{\sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k}$.

Proof (continued). We now compute:

$$\begin{array}{rcll} q^n & = & |\Sigma^n| & \text{because } |\Sigma| = q \\ & = & |\bigcup_{i=1}^{m} B_{d-1}(\mathbf{c}_i)| & \text{by the Claim} \\ & \leq & \sum_{i=1}^{m} |B_{d-1}(\mathbf{c}_i)| & \\ & = & m \sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k & \text{by Proposition 3.1} \\ & = & |C| \sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k & \text{because } m = |C| \end{array}$$

It follows that $|C| \geq \frac{q^n}{\sum_{k=0}^{d-1} \binom{n}{k}(q-1)^k}$, which is what we needed to show.