Error-correcting codes and applications

Klara Stokes

November 20, 2017

Summary and notation

Consider

- $\mathbb{F}_q$: a finite field (if $q = 2$, then $\mathbb{F}_q$ are the binary numbers),
- $V = V(\mathbb{F}_q, n)$: a vector space over $\mathbb{F}_q$ of dimension n (so vectors have length n),
- $U = U(\mathbb{F}_q, k)$: a vector space over $\mathbb{F}_q$ of dimension k (so vectors have length k),
- G: a $k \times n$ matrix over $\mathbb{F}_q$ of rank k.

The set of vectors (codewords) $C = \{uG : u \in U\} \subseteq V$
forms a **linear error correcting code**.

Summary and notation

Consider

- $\mathbb{F}_q$: a finite field (if $q = 2$, then $\mathbb{F}_q$ are the binary numbers),
- $V = V(\mathbb{F}_q, n)$: a vector space over $\mathbb{F}_q$ of dimension n (so vectors have length n),
- $U = U(\mathbb{F}_q, k)$: a vector space over $\mathbb{F}_q$ of dimension k (so vectors have length k),
- G: a $k \times n$ matrix over $\mathbb{F}_q$ of rank k.

The set of vectors (codewords) $C = \{uG : u \in U\} \subseteq V$
forms a **linear error correcting code**.

The code has

- alphabet $\mathbb{F}_q$
- length n,
- dimension k,
- information rate $\frac{k}{n}$,
- generator matrix G.

Summary and notation

The code $C = \{uG : u \in U\} \subseteq V$ is the image of the vector space U under the linear transformation defined by G: $T_G : U \to V$.

Note that C is a subspace of V of dimension k (same as U).

The vector space U is the set of vectors representing information.

The codes minimum distance d is the minimum (Hamming) distance between any two codewords.

To correct many errors the minimum distance d should be large.

Constructing a good code therefore means selecting vectors far away from each other.

Applications of linear error-correcting codes

Examples of different functionalities of linear block codes:

- **Forward error correction (channel coding)**
- **Cryptography (McEliece-type cryptosystems)**
- Group testing (compressed sensing and identifying codes)

1. Forward error correction/channel coding
   - General overview
   - Data storage
   - Pseudonymization in medical research

2. Cryptography
   - Reminder of public key cryptography
   - Code-based public-key cryptography

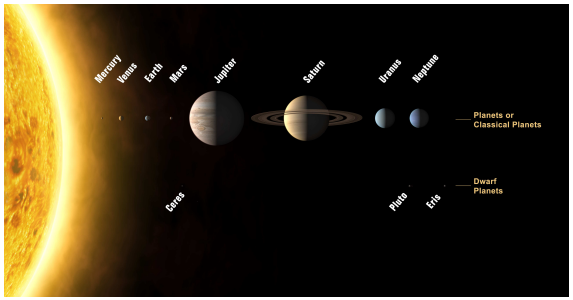
Forward error correction

Information is transmitted over a noisy channel.

There are applications where **resending lost information is expensive or impossible.**

Then we use **forward error correction** or **channel coding**.

Examples:

- 

  Space communications.
  When we notice data is lost it is too late to resend.

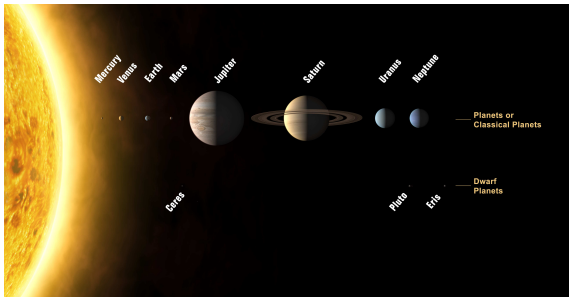
Forward error correction

Information is transmitted over a noisy channel.

There are applications where **resending lost information is expensive or impossible.**

Then we use **forward error correction** or **channel coding**.

Examples:

- 

  Space communications.
  When we notice data is lost it is too late to resend.
- Wireless communications
  When we notice data is lost we already need the next data block.

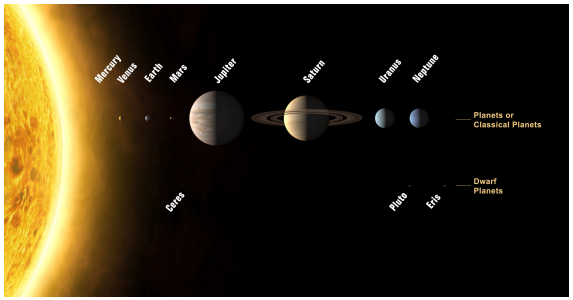
Forward error correction

Information is transmitted over a noisy channel.

There are applications where **resending lost information is expensive or impossible.**

Then we use **forward error correction** or **channel coding**.

Examples:

- 

  Space communications.
  When we notice data is lost it is too late to resend.
- Wireless communications
  When we notice data is lost we already need the next data block.
- Data storage
  When we notice data is lost there is no other copy to recover it from.

Codes for forward error correction: design criteria

What is important to consider?

- The channel (erasures/errors, fading, error-rate, etc).
- Accepted delay (voice conversation: short delay /email: long delay).
- Computing resources (how long time will coding/decoding take).

From this, make choices of code parameters like:

- Alphabet (typically: a finite field F_q where $q = 2^n$).
- How many errors to correct in each block?
- Block length.
- Size of codebook (number of codewords).
- Available coding/decoding algorithms.

Typical research questions

So what kind of questions is interesting when studying forward error correction?

- Give upper bounds for:
  - Amount of information that can be transmitted over the channel (e.g. Shannon limit).
  - Number of codewords of code given error-correcting capacity and length of codewords (e.g. Singleton bound: $d \leq n - k + 1$).
- Construct codes approaching/attaining these bound.
- Construct fast coding/decoding algorithms.

Main tools: combinatorics, geometry and linear algebra.

1. Forward error correction/channel coding
   - General overview
   - Data storage
   - Pseudonymization in medical research

2. Cryptography
   - Reminder of public key cryptography
   - Code-based public-key cryptography

Data storage (one disk)

When data is stored (on disks, tapes, etc), the channel is the storage medium.

Example.
Error-correction for compact discs (CD) often uses Reed-Solomon codes.

Reed-Solomon codes are MDS codes.

MDS codes are codes which attain the Singleton bound $d \leq n - k + 1$, so they are optimal in some sense.

Example.
Error-correction for Micro-SD cards often uses BCH codes.

BCH codes are cyclic codes: it is easy to prescribe error-correcting capacity and decoding is easy.

Data storage (multiple disks)

Modern disks are **cheap** but **error prone**.

Solution: use n disks and store n copies of data.

That's a repetition code!

Every symbol is repeated n times.

Information rate: $\frac{1}{n}$.

We can do better by using, for example, an MDS-code.

Storage codes: general setting

Linear error-correcting code of length n.

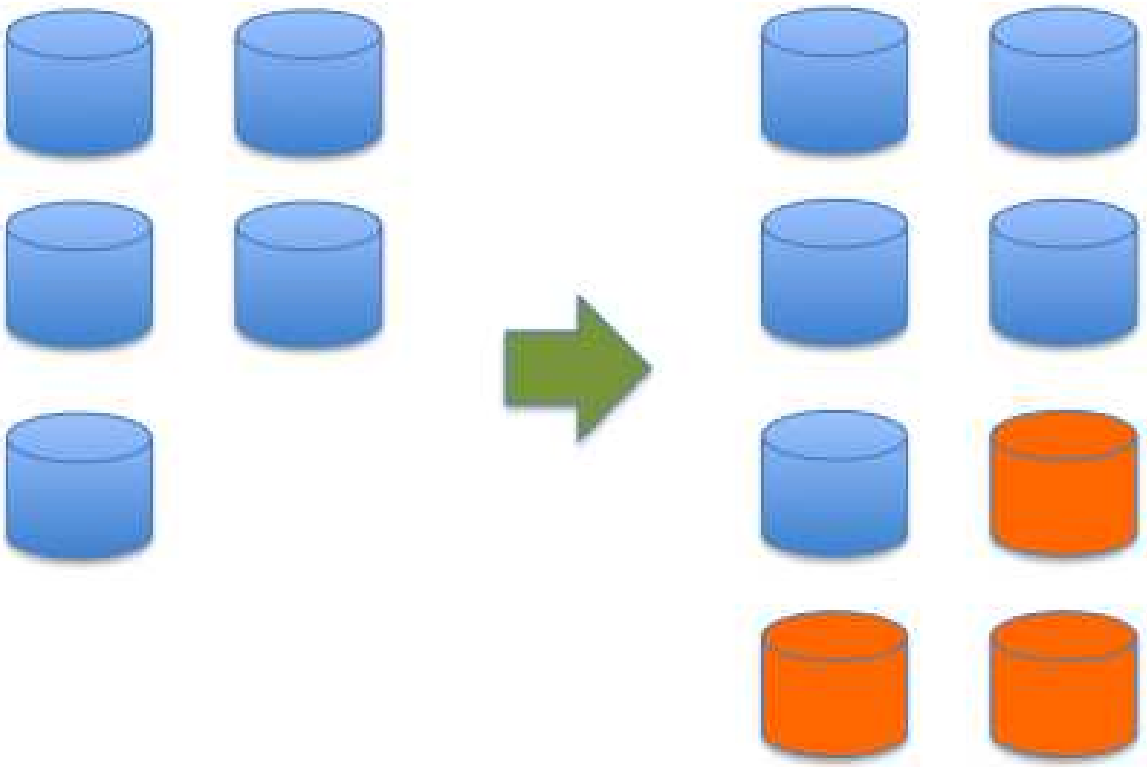
Assign one symbol of the codeword to each of the n disk.

Suppose one disk fails: an error in the codeword.

If the code can correct e errors, then data will survive e disk failures.

MDS-codes attain the Singleton bound: $d \leq n - k + 1$.

They can correct $\frac{n-k}{2}$ errors or $n - k$ erasures (i.e. when you know where the error is).

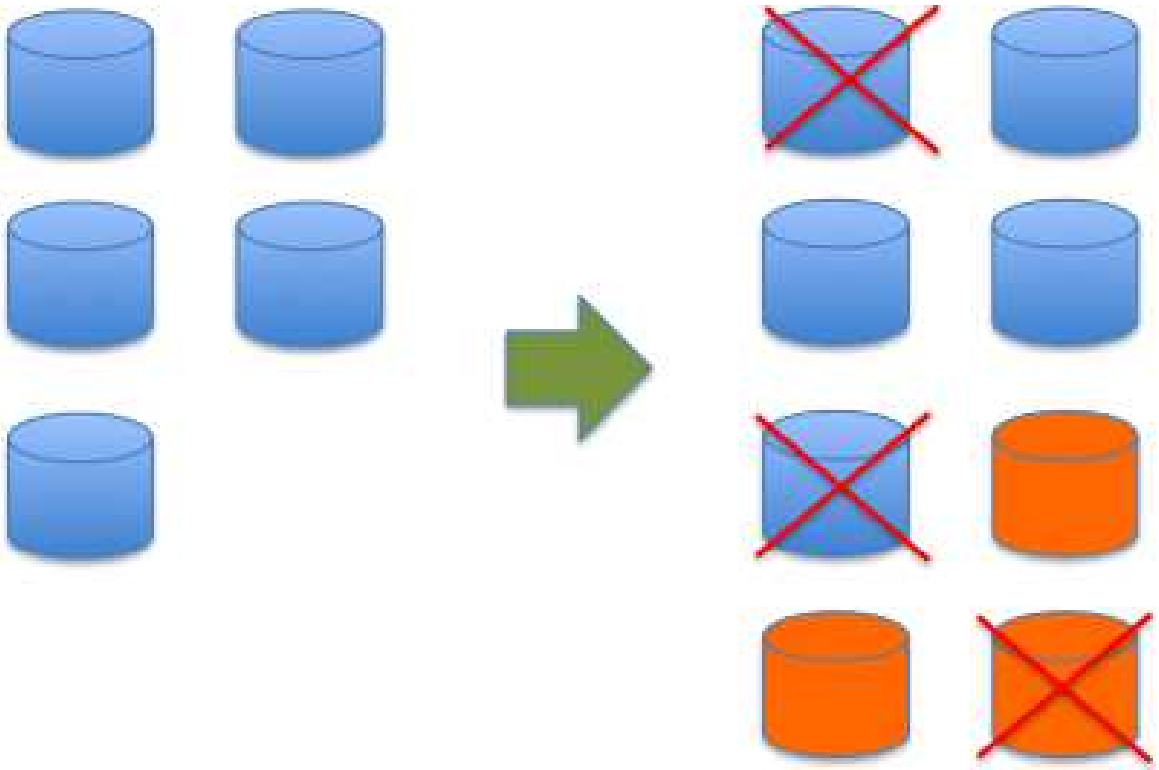


An MDS-code of dimension $k = 5$ and length $n = 8$ have minimum distance $d = n - k + 1 = 8 - 5 + 1 = 4$.

It corrects $n - k = 3$ erasures.

MDS-codes attain the Singleton bound: $d \leq n - k + 1$.

It can correct $\frac{n-k}{2}$ errors or $n - k$ erasures (i.e. when you know where the error is).

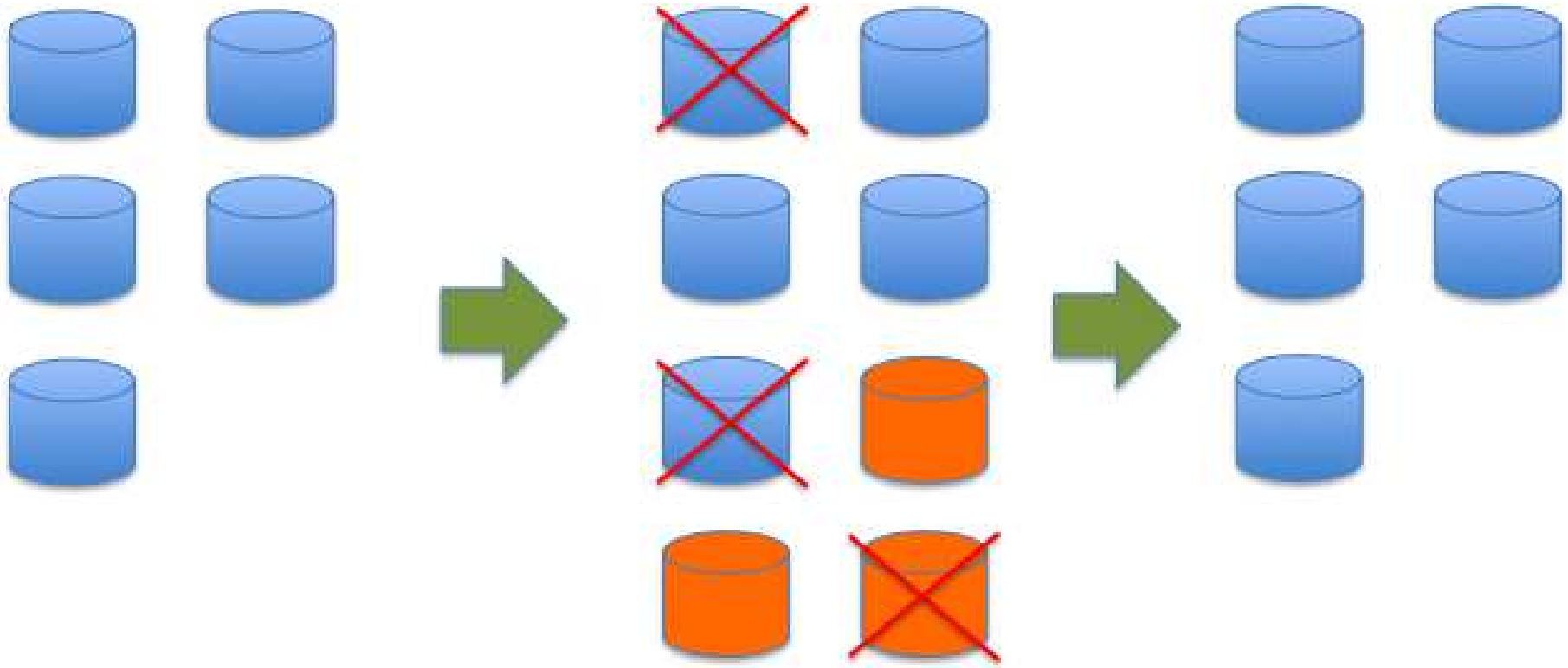


An MDS-code of dimension $k = 5$ and length $n = 8$ have minimum distance $d = n - k + 1 = 8 - 5 + 1 = 4$.

It corrects $n - k = 3$ erasures.

MDS-codes attain the Singleton bound: $d \leq n - k + 1$.

It can correct $\frac{n-k}{2}$ errors or $n - k$ erasures (i.e. when you know where the error is).



An MDS-code of dimension $k = 5$ and length $n = 8$ have minimum distance $d = n - k + 1 = 8 - 5 + 1 = 4$.

It corrects $n - k = 3$ erasures.

Storage codes: other functionalities

In the cloud the cloud provider keeps track of failing disks.

Before e disk failures occur, they have to recode the information so it is not lost.

Downloading the coded information, decode, recode and upload is inefficient.

Solution: Regenerating codes.

Regenerating codes allow a lost symbol/disk to repair itself by contacting a small number of neighboring symbols/disks.

1. Forward error correction/channel coding
   - General overview
   - Data storage
   - Pseudonymization in medical research

2. Cryptography
   - Reminder of public key cryptography
   - Code-based public-key cryptography

Why pseudonymization?

Laws and regulations require data collected in distinct medical research projects to be **pseudonymized** before combined.

A pseudonym is an identifier used to keep track of the data of an individual.

A pseudonym is different from a name or a personal number, in that it should give no reference to the individual (my personal number is almost public information).

Example: German PIDs

German Federal Data Protection Act:

- Replace identifying patient characteristics by a label to preclude identification of a patient by unauthorized persons.
- Clinical trials: Names, initials, birth dates are not allowed.
- Medicinal Products Act: Only pseudonymized data may be passed to the sponsor.

Example: German PIDs

Design challenges of a patient identifier (PID):

- Distinguish 1 billion individuals by a unique PID.
- Medical personal are in a hurry and make lots of typos. Consequences:
  - False classification of patients.
  - Possible severe consequences in diagnosis and therapy.
  - Increasing variance in statistical analyses.

Solution: error-correcting pseudonyms (PID)!

Example: German PIDs

German researcher suggest a $32 - [8, 6, 3]$ MDS-code:

- Minimum distance 3.
- Optimal code (attains the Singleton bound).
- Detects 2 errors.
- Corrects 1 error.
- Corrects transposition of adjacent distinct characters.

Homework: This is forward error-correction. What is the channel in this application?

1. Forward error correction/channel coding
   - General overview
   - Data storage
   - Pseudonymization in medical research

2. Cryptography
   - Reminder of public key cryptography
   - Code-based public-key cryptography

Assymmetric/public key cryptography

Secrecy: Alice **encrypts** (transforms) the message into something only Bob can **decrypt** (read).

Symmetric key cryptography: Alice and Bob use the **same key** for encryption and decryption.

Assymmetric key cryptography: Alice and Bob have **different keys**.

- The **public key** is for encryption (Alice can get it for example from Bob's homepage),
- The **secret key** is for decrypting (Bob keeps it secret).

Calculating the secret key from the public key must be a

COMPUTATIONALLY HARD PROBLEM.

Example: RSA is based on the *hard problem* of integer factorization.

(Integer factorization is not so hard if you have a quantum computer... and it is not believed to be NP-complete.)

1. Forward error correction/channel coding
   - General overview
   - Data storage
   - Pseudonymization in medical research

2. Cryptography
   - Reminder of public key cryptography
   - Code-based public-key cryptography

A hard problem in coding theory

The decoding problem: given a received vector v and a code C, find the closest codeword $c \in C$.

The general decoding problem of a linear code is NP-complete.

In other words: the existence of a polynomial time algorithm for decoding any linear code would imply that $P = NP$. (The most important problem in computer science?)

There is an algorithm (of exponential complexity) which decodes a received vector v to the correct codeword by searching a set of **exponentially many (in terms of the length of v)** candidate codewords.

If $P \neq NP$, then we can't do much better!

Decoding algorithms for special codes

If all decoding algorithms were of exponential complexity, coding theory would be pretty useless.

There are many efficient (polynomial time) algorithms for decoding.

A given family of codes becomes interesting because of the coding and decoding algorithms they feature.

It is the **algebraic and geometric structure** of the code which decides what algorithms are useful.

Example.
Binary Goppa codes can be decoded using the Patterson algorithm.

McEliece cryptosystem

Public key: A matrix and an integer $(\hat{G}, t)$.

- The matrix $\hat{G}$ is the generator matrix of a linear code $\hat{C}$ obtained from a generator matrix G of a **Goppa code** C as $\hat{G} = SGP$ where S and P are certain randomly chosen invertible matrices.
- t is the number of errors that C (and $\hat{C}$) can correct.

We can regard $\hat{G}$ as a "scrambled" generator matrix of the generator matrix G of the Goppa code C.

McEliece cryptosystem

Public key: A matrix and an integer $(\hat{G}, t)$.

- The matrix $\hat{G}$ is the generator matrix of a linear code $\hat{C}$ obtained from a generator matrix G of a **Goppa code** C as $\hat{G} = SGP$ where S and P are certain randomly chosen invertible matrices.
- t is the number of errors that C (and $\hat{C}$) can correct.

We can regard $\hat{G}$ as a "scrambled" generator matrix of the generator matrix G of the Goppa code C.

Private key: The three matrices (S, G, P).

Bob encodes the message vector m as $x = m\hat{G} + e_t$, where e_t is an error vector with t errors.

Alice has the matrices S, P which scrambled G to $\hat{G}$. They define invertible linear transformations T_S and T_P.

She uses the inverse transformations to allow the Patterson algorithm to remove the error e_t from $T_P^{-1}(x)$, and then recover m by applying T_S^{-1}.

McEliece cryptosystem

McEliece cryptosystem was one of the first non-deterministic cryptosystems.

Advantage: Immune to Shor's algorithm (which makes integer factorization a not-so-hard problem for quantum computers.)

Drawback: Large key (post-quantum: about 1 MB).

Thank you!