An Introduction to (Network) Coding Theory

Anna-Lena Horlemann-Trautmann

University of St. Gallen, Switzerland

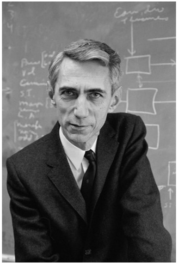
April 24th, 2018

Outline

1. Coding Theory
   - Introduction
   - Reed-Solomon Codes
2. Network Coding Theory
   - Introduction
   - Gabidulin Codes
3. Summary and Outlook

– A little bit of history –

2016 was the 100th anniversary of the
Father of Information Theory



Claude Shannon (1916 - 2001)[1]

[1] picture from www.techzibits.com

Shannon's pioneering works in information theory:

- Channel coding (1948):
  - Noisy-channel coding theorem/Shannon capacity (maximum information transfer rate for a given channel and noise level)
- Compression (1948):
  - Source coding theorem (limits to possible data compression)
- Cryptography (1949):
  - One-time pad is the only theoretically unbreakable cipher

Shannon provided answers to questions of the type

"What is possible in theory?"

Subsequent research:

- how to algorithmically achieve those optimal scenarios
- other types of channels
- lossy compression
- computationally secure cryptography

Channel Coding

... deals with noisy transmission of information

- over space (communication)
- over time (storage)

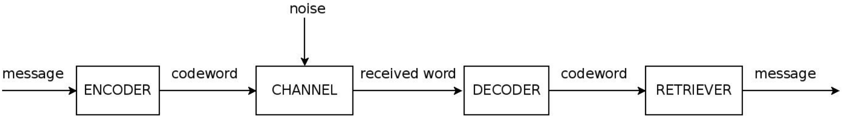
Channel Coding

... deals with noisy transmission of information

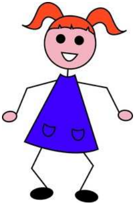
- over space (communication)
- over time (storage)

To deal with the noise

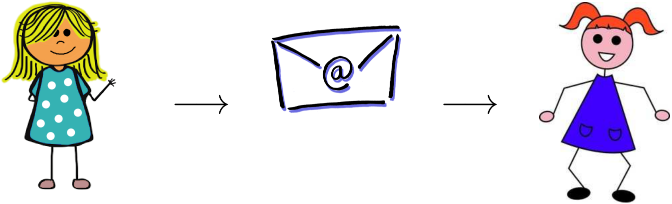
- the data is *encoded* with added redundancy,
- the receiver can "filter out" the noise (*decoding*)
- and then *recover* the sent data.



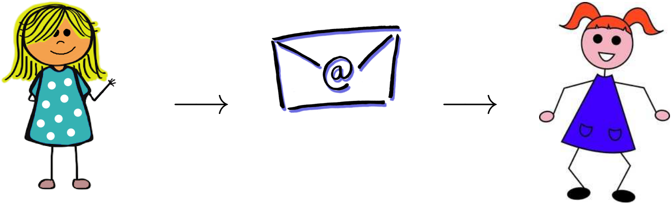
Classical channel coding:



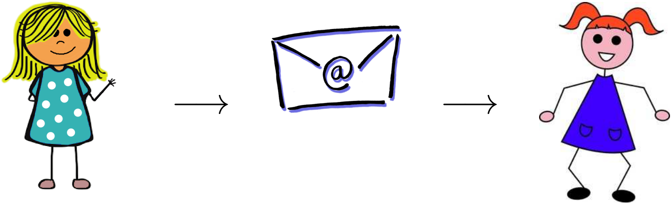
Classical channel coding:



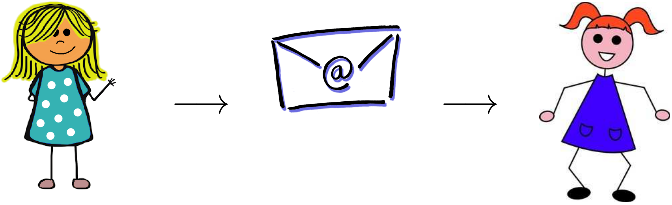
Classical channel coding:

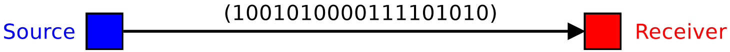


Classical channel coding:

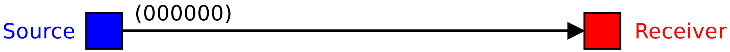


Classical channel coding:



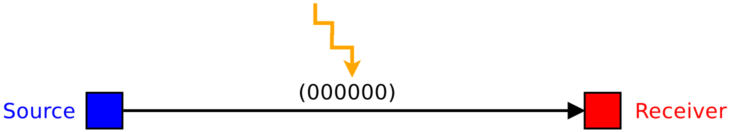


Classical channel coding:



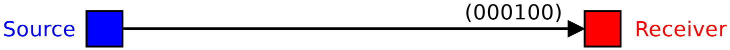
yes = (111111)
no = (000000)

Classical channel coding:



yes = (111111)
no = (000000)

Classical channel coding:



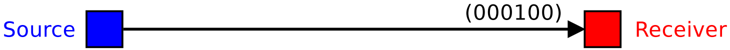
yes = (111111)
no = (000000)

Classical channel coding:

yes = (111111)
no = (000000)

Receiver: (000100) is *closer* to (000000) than to (111111)

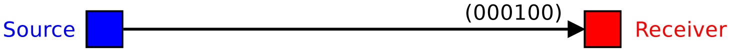
Classical channel coding:



yes = (111111)
no = (000000)

Receiver: (000100) is *closer* to (000000) than to (111111)
$\Longrightarrow$ decode to (000000) = no

Classical channel coding:

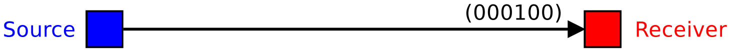


yes = (111111)
no = (000000)

Receiver: (000100) is *closer* to (000000) than to (111111)
$\Longrightarrow$ decode to (000000) = no

- The closeness can be measured by the *Hamming metric*.

Classical channel coding:

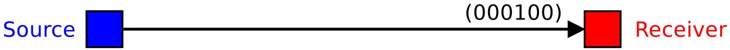


yes = (111111)
no = (000000)

Receiver: (000100) is *closer* to (000000) than to (111111)
$\Longrightarrow$ decode to (000000) = no

- The closeness can be measured by the *Hamming metric*.
- The larger the distance between the codewords, the more errors can be corrected.

Classical channel coding:



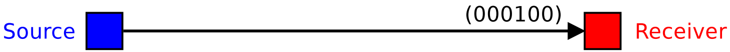
yes = (111111)
no = (000000)

Receiver: (000100) is *closer* to (000000) than to (111111)

$\Longrightarrow$ decode to (000000) = no

- The closeness can be measured by the *Hamming metric*.
- The larger the distance between the codewords, the more errors can be corrected.
- Tradeoff: The longer the codewords, the lower the information transmission rate.

Classical channel coding:



yes = (111111)
no = (000000)

Receiver: (000100) is *closer* to (000000) than to (111111)
$\Longrightarrow$ decode to (000000) = no

- The closeness can be measured by the *Hamming metric*.
- The larger the distance between the codewords, the more errors can be corrected.
- Tradeoff: The longer the codewords, the lower the information transmission rate.

Errors/noise

- Maybe you wonder why the error correction is so important.
- This is because we do not live in a perfect vacuum where everything works “as it should”.
- Noise is around everywhere, think of particles in the air (when sending data wireless), or scratches on a CD (when storing data on the CD), or electromagnetic interference in cables (when sending data over wires).

Errors/noise

- Maybe you wonder why the error correction is so important.
- This is because we do not live in a perfect vacuum where everything works "as it should".
- Noise is around everywhere, think of particles in the air (when sending data wireless), or scratches on a CD (when storing data on the CD), or electromagnetic interference in cables (when sending data over wires).
- However, we always assume that errors are less likely than noise-free transmission (per element). Thus the most likely sent codeword corresponds to the one with the least number of errors, compared to the received word.

Data representation over finite fields

- You have probably heard that computers (or smart phones and similar devices) work with *binary* data.
- However, some technologies like e.g. flash drives also use more numbers than just 0 and 1.
- Even for binary representation it is often advantageous to represent data in binary *extension fields*.
- In general we say that data is represented as vectors over some finite field $\mathbb{F}_q$.

Definition

A *block code* is a subset $C \subseteq \mathbb{F}_q^n$. The *Hamming distance* of $u, v \in \mathbb{F}_q^n$ is defined as

$$d_H((u_1, \ldots, u_n), (v_1, \ldots, v_n)) := |\{i \mid u_i \neq v_i\}|.$$

The *minimum (Hamming) distance* of the code is defined as

$$d_H(C) := \min\{d_H(u, v) \mid u, v \in C, u \neq v\}.$$

The *transmission rate* of C is defined as $\log_q(|C|)/n$.

Theorem

Let C be a code with minimum Hamming distance $d_H(C) = d$. Then for any codeword $c \in C$ any $(d_H(C) - 1)/2$ errors can be corrected.

$\implies$ the *error correction capability* of C is $\lfloor (d_H(C) - 1)/2 \rfloor$

Example (repetition code):

- Remember the code from the introduction slides:

$$C = \{(000000), (111111)\}$$

  This code has transmission rate $\log_2(2)/6 = 1/6$.
- This code has minimum Hamming distance 6 (since all coordinates differ).
- The error correction capability is $\lfloor (6-1)/2 \rfloor = 2$.
- Indeed, if we receive e.g. (110000), the unique closest codeword is (000000).
- However, for (111000) there is no unique closest codeword, hence we cannot correct 3 errors.

The general repetition code:

Definition

The *repetition code* over $\mathbb{F}_q$ of length n is defined as

$$C := \{\underbrace{(x, x, \ldots, x)}_{n} \mid x \in \mathbb{F}_q\}.$$

It has cardinality q and minimum Hamming distance n.

The general repetition code:

Definition

The *repetition code* over $\mathbb{F}_q$ of length n is defined as

$$C := \{\underbrace{(x, x, \dots, x)}_{n} \mid x \in \mathbb{F}_q\}.$$

It has cardinality q and minimum Hamming distance n.

- transmission rate $= 1/n$
- error correction capability $= \lfloor (n-1)/2 \rfloor$

Typical questions in channel coding theory:

- For a given error correction capability, what is the best transmission rate?
  $\implies$ packing problem in metric space $(\mathbb{F}_q^n, d_H)$
- How can one efficiently encode, decode, recover the messages?
  $\implies$ algebraic structure in the code
- What is the trade-off between the two above?

Typical questions in channel coding theory:

- For a given error correction capability, what is the best transmission rate?
  $\Longrightarrow$ packing problem in metric space $(\mathbb{F}_q^n, d_H)$
- How can one efficiently encode, decode, recover the messages?
  $\Longrightarrow$ algebraic structure in the code
- What is the trade-off between the two above?

Typical tools used in classical setup:

- linear subspaces of $\mathbb{F}_q^n$
- polynomials (and their roots) in $\mathbb{F}_q[x]$
- finite projective geometry

The most prominent family of error-correcting codes

–

Reed-Solomon codes

Definition (Reed-Solomon codes)

Let $a_1, \ldots, a_n \in \mathbb{F}_q$ be distinct. The code

$$C = \{(f(a_1), f(a_2), \ldots, f(a_n)) \mid f \in \mathbb{F}_q[x], \deg f < k\}$$

is called a *Reed-Solomon code* of length n and dimension k. It has minimum Hamming distance $n - k + 1$ (optimal).

Definition (Reed-Solomon codes)

Let $a_1, \dots, a_n \in \mathbb{F}_q$ be distinct. The code

$$C = \{(f(a_1), f(a_2), \dots, f(a_n)) \mid f \in \mathbb{F}_q[x], \deg f < k\}$$

is called a *Reed-Solomon code* of length n and dimension k. It has minimum Hamming distance $n - k + 1$ (optimal).

A Reed-Solomon code is a linear subspace of $\mathbb{F}_q^n$ of dimension k, it can be represented by a (row) generator matrix

$$G = \begin{pmatrix} 1 & 1 & \dots & 1 \\ a_1 & a_2 & \dots & a_n \\ a_1^2 & a_2^2 & \dots & a_n^2 \\ \vdots & & & \vdots \\ a_1^{k-1} & a_2^{k-1} & \dots & a_n^{k-1} \end{pmatrix}.$$

Example:

- Consider $\mathbb{F}_3 = \{0, 1, 2\}$, $n = 3, k = 2$ and the evaluation points $a_1 = 0, a_2 = 1, a_3 = 2$.
- Polynomials of degree ≤ 0: $\quad 0, 1, 2$
- Polynomials of degree 1: $\quad x, x+1, x+2, 2x, 2x+1, 2x+2$
- Codewords:

$f(x)$	$(f(0), f(1), f(2))$
0	(000)
1	(111)
2	(222)
x	(012)
$x+1$	(120)
$x+2$	(201)
$2x$	(021)
$2x+1$	(102)
$2x+2$	(210)

$f(x)$	$(f(0), f(1), f(2))$
0	(000)
1	(111)
2	(222)
x	(012)
$x+1$	(120)
$x+2$	(201)
$2x$	(021)
$2x+1$	(102)
$2x+2$	(210)

The generator matrix in reduced row echelon form of this code is

$$G = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \end{pmatrix}.$$

$f(x)$	$(f(0), f(1), f(2))$
0	(000)
1	(111)
2	(222)
x	(012)
$x+1$	(120)
$x+2$	(201)
$2x$	(021)
$2x+1$	(102)
$2x+2$	(210)

The generator matrix in reduced row echelon form of this code is

$$G = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \end{pmatrix}.$$

$\Longrightarrow$ any two words differ in $\geq n-k+1 = 3-2+1 = 2$ positions ($d_H(C) = 2$).

Why Reed-Solomon codes are awesome:

- One can show that for a linear code of dimension k and length n, the minimum Hamming distance cannot exceed $n - k + 1$ (Singleton bound).
  $\implies$ RS-codes are optimal, since they reach this bound.

Why Reed-Solomon codes are awesome:

- One can show that for a linear code of dimension k and length n, the minimum Hamming distance cannot exceed $n-k+1$ (Singleton bound).
  $\Longrightarrow$ RS-codes are optimal, since they reach this bound.
- Decoding can be translated into a polynomial interpolation problem.
  $\Longrightarrow$ RS-codes can be decoded quite efficiently.

Why Reed-Solomon codes are awesome:

- One can show that for a linear code of dimension k and length n, the minimum Hamming distance cannot exceed $n - k + 1$ (Singleton bound).
  $\Longrightarrow$ RS-codes are optimal, since they reach this bound.
- Decoding can be translated into a polynomial interpolation problem.
  $\Longrightarrow$ RS-codes can be decoded quite efficiently.

Why RS-codes are not the solution to everything:

- The underlying field size needs to be as large as the length!

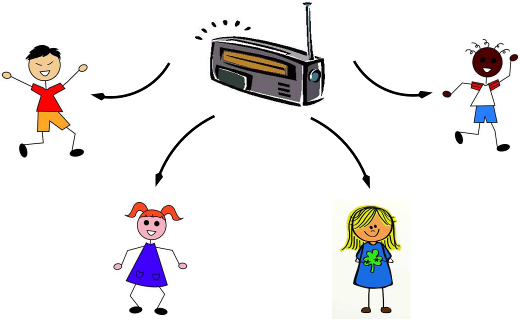
1 Coding Theory
- Introduction
- Reed-Solomon Codes

2 Network Coding Theory
- Introduction
- Gabidulin Codes

3 Summary and Outlook

Network channel

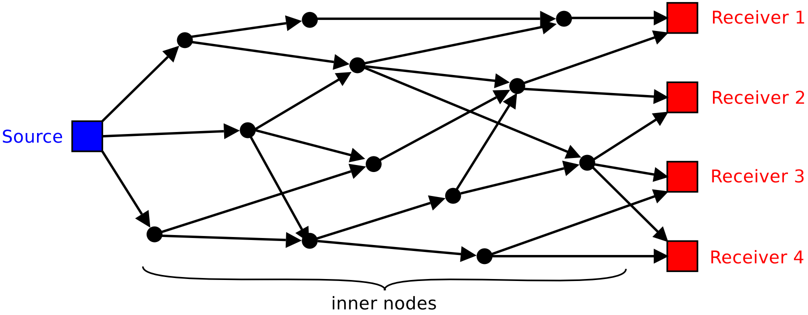
The multicast model:



All receivers want to get the same information at the same time.

Network channel

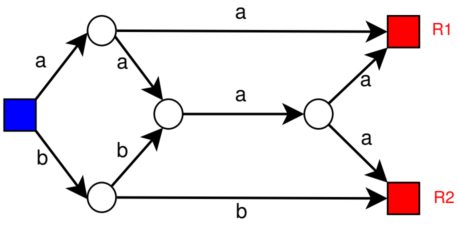
The multicast model:



All receivers want to get the same information at the same time.

Example (Butterfly Network)

Linearly combining is better than forwarding:

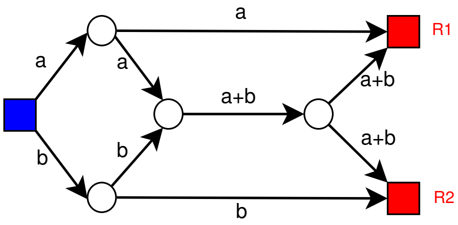


R1 receives only a, R2 receives a and b.

- Forwarding: need 2 transmissions to transmit a, b to both receivers

Example (Butterfly Network)

Linearly combining is better than forwarding:



R1 and R2 can both recover a and b with one operation.

- Forwarding: need 2 transmissions to transmit a, b to both receivers
- Linearly combining: need 1 transmission to transmit a, b to both receivers

It turns out that linear combinations at the inner nodes are "sufficient" to reach capacity:

Theorem

One can reach the capacity of a single-source multicast network channel with linear combinations at the inner nodes.

It turns out that linear combinations at the inner nodes are "sufficient" to reach capacity:

Theorem

One can reach the capacity of a single-source multicast network channel with linear combinations at the inner nodes.

When we consider large or time-varying networks, we allow the inner nodes to transmit *random linear combinations* of their incoming vectors.

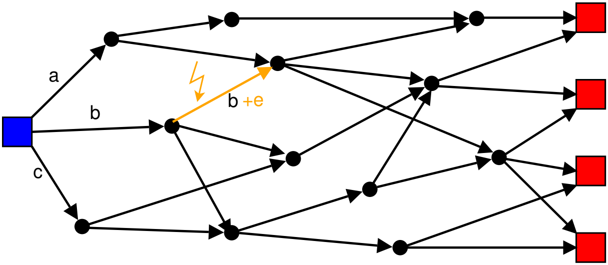
Theorem

One can reach the capacity of a single-source multicast network channel with random linear combinations at the inner nodes, provided that the field size is large.

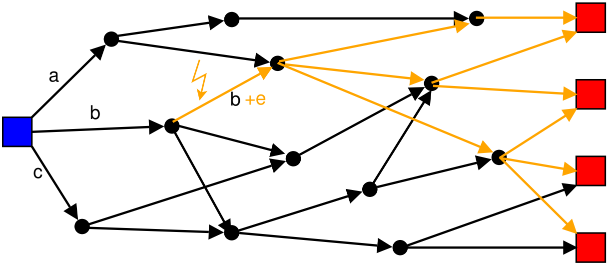
Two settings for *linear* network coding:

- *Coherent* (linear) network coding – we prescribe each inner node the linear transformation
- *Non-coherent* or *random* (linear) network coding
    - e.g. time-varying networks, large networks, ...
    - allow each inner node to send out a random linear combination of its incoming vectors

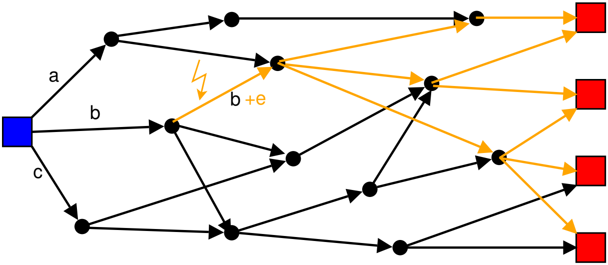
Problem 1: errors propagate!



Problem 1: errors propagate!

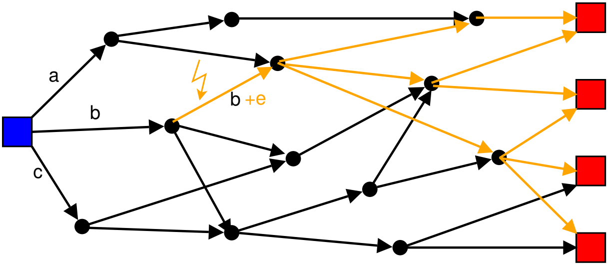


Problem 1: errors propagate!



Problem 2: receiver does not know the random operations (in non-coherent setting)

Problem 1: errors propagate!



Problem 2: receiver does not know the random operations (in non-coherent setting)

Solution: Use a metric space such that

1. # of errors is reflected in the distance between points, and
2. the points are invariant under linear combinations (for non-coherent).

For the coherent case:

Definition

- matrix space: $\mathbb{F}_q^{m\times n}$
- rank distance: $d_R(U,V) := \mathrm{rank}(U-V)$

$\mathbb{F}_q^{m\times n}$ equipped with d_R is a metric space.

Definition

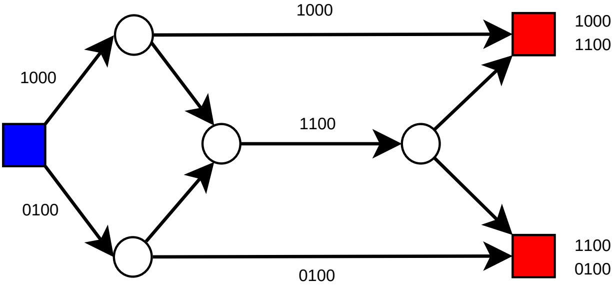
A *rank-metric code* is a subset of $\mathbb{F}^{m\times n}$. The *minimum rank distance* of the code $C \subseteq \mathbb{F}^{m\times n}$ is defined as

$$d_R(C) := \min\{d_R(U,V) \mid U,V \in C, U \neq V\}.$$

A rank-metric code C can correct any error (matrix) of rank at most $(d_R(C)-1)/2$.

Example (in $\mathbb{F}_2^{2\times 4}$)

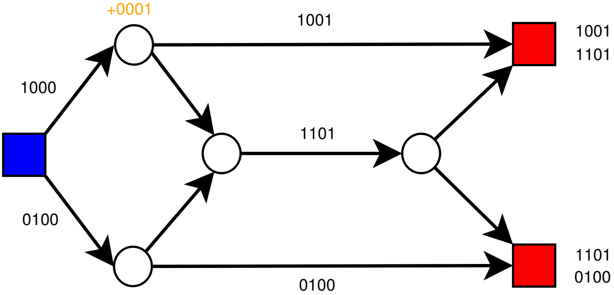
$$C = \left\{ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \right\}, d_R(C) = 2.$$



No errors: receive $\underbrace{\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}}_{A_1} \cdot$sent, respectively $\underbrace{\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}}_{A_2} \cdot$sent

Example (in $\mathbb{F}_2^{2\times 4}$)

$$C = \left\{ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \right\}, d_R(C) = 2.$$



One error:
$d_R(A_i^{-1} \cdot \text{received, sent}) = 1, d_R(A_i^{-1} \cdot \text{received, other}) = 2$

For the non-coherent case:

Definition

- Grassmann variety: $\mathcal{G}_q(k,n) := \{U \leq \mathbb{F}_q^n \mid \dim(U) = k\}$
- subspace distance: $d_S(U,V) := 2k - 2\dim(U \cap V)$

$\mathcal{G}_q(k,n)$ equipped with d_S is a metric space.

Definition

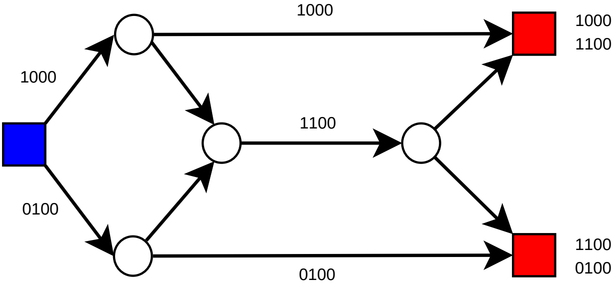
A *(constant dimension) subspace code* is a subset of $\mathcal{G}_q(k,n)$.
The *minimum distance* of the code $C \subseteq \mathcal{G}_q(k,n)$ is defined as

$$d_S(C) := \min\{d_S(U,V) \mid U,V \in C, U \neq V\}.$$

The error-correction capability in the network coding setting of a subspace code C is $(d_S(C)-1)/2$.

Example (in $\mathcal{G}_2(2,4)$)

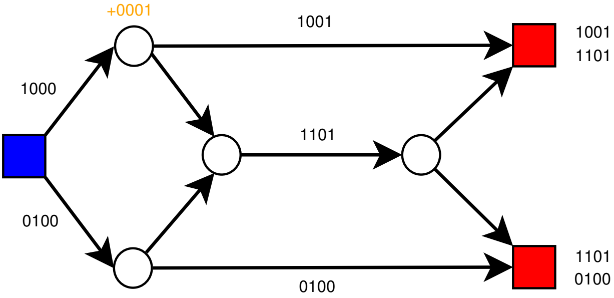
$$C = \left\{ \mathrm{rs}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \mathrm{rs}\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \right\}, d_S(C) = 4.$$



No errors: receive a (different) basis of the same vector space

Example (in $\mathcal{G}_2(2,4)$)

$$C = \left\{ \mathrm{rs}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \mathrm{rs}\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \right\}, d_S(C) = 4.$$



One error: $d_S(\text{received, sent}) = 2, d_S(\text{received, other}) = 4$

Research goals

- Find good packings in $(\mathbb{F}_q^{m\times n}, d_R)$, respectively $(\mathcal{G}_q(k,n), d_S)$.
  $\implies$ best transmission rate for given error correction capability
- Find good packings in $\mathbb{F}_q^{m\times n}$, respectively $\mathcal{G}_q(k,n)$, with algebraic structure.
  $\implies$ good encoding/decoding algorithms

Research goals

- Find good packings in $(\mathbb{F}_q^{m\times n}, d_R)$, respectively $(\mathcal{G}_q(k,n), d_S)$.
  $\implies$ best transmission rate for given error correction capability
- Find good packings in $\mathbb{F}_q^{m\times n}$, respectively $\mathcal{G}_q(k,n)$, with algebraic structure.
  $\implies$ good encoding/decoding algorithms

Typical tools

- linearized polynomials in $\mathbb{F}_q[x]$
- Singer cycles, difference sets
- (partial) spreads

The most prominent family of rank-metric codes

–

Gabidulin codes

Preliminaries:

- Isomorphism:

$$\mathbb{F}_{q^m} \cong \mathbb{F}_q^m$$

- This induces another isomorphism:

$$\mathbb{F}_{q^m}^n \cong \mathbb{F}_q^{m \times n}$$

Preliminaries:

- Isomorphism:

$$\mathbb{F}_{q^m} \cong \mathbb{F}_q^m$$

- This induces another isomorphism:

$$\mathbb{F}_{q^m}^n \cong \mathbb{F}_q^{m \times n}$$

- Linearized polynomial:

$$f(x) = \sum_{i=0}^{d} f_i x^{q^i}$$

- The set of all linearized polynomials is denoted by $\mathcal{L}_q[x]$.

Definition (Gabidulin codes)

Let $a_1, \ldots, a_n \in \mathbb{F}_{q^m}$ be linearly independent over $\mathbb{F}_q$. The code

$$C = \{(f(a_1), f(a_2), \ldots, f(a_n)) \mid f \in \mathcal{L}_q[x], \deg f < q^k\}$$

is called a *Gabidulin code* of length n and dimension k. It has minimum rank distance $n - k + 1$ (optimal).

Definition (Gabidulin codes)

Let $a_1, \dots, a_n \in \mathbb{F}_{q^m}$ be linearly independent over $\mathbb{F}_q$. The code

$$C = \{(f(a_1), f(a_2), \dots, f(a_n)) \mid f \in \mathcal{L}_q[x], \deg f < q^k\}$$

is called a *Gabidulin code* of length n and dimension k. It has minimum rank distance $n - k + 1$ (optimal).

A Gabidulin code is a linear subspace of $\mathbb{F}_{q^m}^n$ of dimension k, it can be represented by a (row) generator matrix

$$G = \begin{pmatrix} a_1 & a_2 & \dots & a_n \\ a_1^q & a_2^q & \dots & a_n^q \\ a_1^{q^2} & a_2^{q^2} & \dots & a_n^{q^2} \\ \vdots & & & \vdots \\ a_1^{q^{k-1}} & a_2^{q^{k-1}} & \dots & a_n^{q^{k-1}} \end{pmatrix}.$$

Example:

- Consider $\mathbb{F}_4 = \{0, 1, \alpha, \alpha+1\}$, $n = 2, k = 1$ and the evaluation points $a_1 = 1, a_2 = \alpha$.
- Lin. polynomials of degree $\leq q^0$: $\quad 0, x, \alpha x, (\alpha+1)x$
- Codewords:

$f(x)$	$(f(1), f(\alpha))$	matrix
0	$(0, 0)$	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$
x	$(1, \alpha)$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
αx	$(\alpha, \alpha+1)$	$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$
$(\alpha+1)x$	$(\alpha+1, 1)$	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$

$f(x)$	$(f(1), f(\alpha))$	matrix
0	$(0,0)$	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$
x	$(1,\alpha)$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
αx	$(\alpha, \alpha+1)$	$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$
$(\alpha+1)x$	$(\alpha+1, 1)$	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$

The generator matrix in reduced row echelon form of this code is

$$G = \begin{pmatrix} 1 & \alpha \end{pmatrix}.$$

$f(x)$	$(f(1), f(\alpha))$	matrix
0	$(0,0)$	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$
x	$(1,\alpha)$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
αx	$(\alpha, \alpha+1)$	$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$
$(\alpha+1)x$	$(\alpha+1, 1)$	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$

The generator matrix in reduced row echelon form of this code is

$$G = \begin{pmatrix} 1 & \alpha \end{pmatrix}.$$

$\implies$ The difference of any two words has full rank: $d_R(C) = 2$.

Why Gabidulin codes are awesome:

- One can show that for a linear rank-metric code of dimension k and size $m \times n$, the minimum rank distance cannot exceed $\max(n, m)(\min(n, m) - k + 1)$ (Singleton-like bound).
  $\Longrightarrow$ Gabidulin codes are optimal, since they reach this bound.

Why Gabidulin codes are awesome:

- One can show that for a linear rank-metric code of dimension k and size $m \times n$, the minimum rank distance cannot exceed $\max(n, m)(\min(n, m) - k + 1)$ (Singleton-like bound).
  $\implies$ Gabidulin codes are optimal, since they reach this bound.
- Decoding can be translated into a linearized polynomial interpolation problem.
  $\implies$ Gabidulin codes can be decoded quite efficiently.

Why Gabidulin codes are awesome:

- One can show that for a linear rank-metric code of dimension k and size $m \times n$, the minimum rank distance cannot exceed $\max(n,m)(\min(n,m) - k + 1)$ (Singleton-like bound).
  $\Longrightarrow$ Gabidulin codes are optimal, since they reach this bound.
- Decoding can be translated into a linearized polynomial interpolation problem.
  $\Longrightarrow$ Gabidulin codes can be decoded quite efficiently.

Difference to RS-codes:

- Although m needs to be at least n, this does not matter much – we can simply transpose the matrices to get a rank-metric code with $m \leq n$.
- Hence, we can construct Gabidulin codes for any q, n, m, k!

How to use Gabidulin codes for the non-coherent setting

Theorem

Let $C \subseteq \mathbb{F}_q^{k\times(n-k)}$ *be a rank-metric code with minimum rank distance* d_R*. Then the lifted code*

$$\mathrm{lift}(C) := \{\mathrm{rs}[I_k \mid U] \mid U \in C\}$$

is a subspace code in $\mathcal{G}_q(k,n)$ *with minimum subspace distance* $d_S = 2d_R$*.*

Theorem

Let $C \subseteq \mathbb{F}_q^{k\times(n-k)}$ be a rank-metric code with minimum rank distance d_R. Then the lifted code

$$\mathrm{lift}(C) := \{\mathrm{rs}[I_k \mid U] \mid U \in C\}$$

is a subspace code in $\mathcal{G}_q(k,n)$ with minimum subspace distance $d_S = 2d_R$.

- Lifted Gabidulin codes are not optimal, but only a factor 4 away from the theoretical upper bound on the cardinality (therefore they are *asymptotically* optimal).
- Decoding the lifted code basically translates to decoding the original rank-metric code.

1. Coding Theory
   - Introduction
   - Reed-Solomon Codes
2. Network Coding Theory
   - Introduction
   - Gabidulin Codes
3. Summary and Outlook

Summary

- We gave an introduction to classical (channel) coding theory.
  - codewords are vectors over finite fields
- The most prominent family of codes for this setup are the Reed-Solomon codes.
- We gave an introduction to network coding theory:
  - coherent (codewords are matrices)
  - non-coherent or random (codewords are subspaces)
- The most prominent family of codes for this setup are the (lifted) Gabidulin codes (also called Reed-Solomon-like codes).

Outlook

- Rank-metric codes (and sometimes subspace codes) are also used in cryptography.
  (Here also non-Gabidulin codes are of interest.)
- Gabidulin codes are also used in distributed storage.
- Other constructions of subspace codes use techniques from
  - projective geometry (spreads, sunflowers)
  - enumerative geometry (intersection numbers)
  - q-analogs of designs (combinatoricss)
  - group theory (orbits in $\mathcal{G}_q(k,n)$) .

Thank you for your attention!
Questions? – Comments?