Erasure codes for distributed storage and related problems

Alexander Barg

University of Maryland, College Park

NASIT, July 2019

THE MAIN MESSAGE OF THIS TUTORIAL:

THE MAIN MESSAGE OF THIS TUTORIAL:

- The task of node repair in distributed storage gives rise to a range of new, previously unrecognized problems in coding theory and related areas of computer science and discrete mathematics.

THE MAIN MESSAGE OF THIS TUTORIAL:

- The task of node repair in distributed storage gives rise to a range of new, previously unrecognized problems in coding theory and related areas of computer science and discrete mathematics.
- These problems have been actively studied for the past decade and led to the emergence of new methods and ideas in these areas.

The main message of this tutorial:

- The task of node repair in distributed storage gives rise to a range of new, previously unrecognized problems in coding theory and related areas of computer science and discrete mathematics.
- These problems have been actively studied for the past decade and led to the emergence of new methods and ideas in these areas.
- The goal of this tutorial is to introduce these methods and the associated results as well as to point out new research directions.

THE MAIN MESSAGE OF THIS TUTORIAL:

- The task of node repair in distributed storage gives rise to a range of new, previously unrecognized problems in coding theory and related areas of computer science and discrete mathematics.
- These problems have been actively studied for the past decade and led to the emergence of new methods and ideas in these areas.
- The goal of this tutorial is to introduce these methods and the associated results as well as to point out new research directions.

ACKNOWLEDGMENTS:

ITZHAK TAMO (Tel Aviv U.)

MIN YE (Princeton U.)

CHEN ZITAN (UMD)

SERGE VLĂDUŢ (U. Marseille)

THE MAIN MESSAGE OF THIS TUTORIAL:

- The task of node repair in distributed storage gives rise to a range of new, previously unrecognized problems in coding theory and related areas of computer science and discrete mathematics.
- These problems have been actively studied for the past decade and led to the emergence of new methods and ideas in these areas.
- The goal of this tutorial is to introduce these methods and the associated results as well as to point out new research directions.

ACKNOWLEDGMENTS:

ITZHAK TAMO (Tel Aviv U.)
MIN YE (Princeton U.)
CHEN ZITAN (UMD)
SERGE VLĂDUŢ (U. Marseille)

THE MAIN MESSAGE OF THIS TUTORIAL:

- The task of node repair in distributed storage gives rise to a range of new, previously unrecognized problems in coding theory and related areas of computer science and discrete mathematics.
- These problems have been actively studied for the past decade and led to the emergence of new methods and ideas in these areas.
- The goal of this tutorial is to introduce these methods and the associated results as well as to point out new research directions.

ACKNOWLEDGMENTS:

ITZHAK TAMO (Tel Aviv U.)

MIN YE (Princeton U.)

CHEN ZITAN (UMD)

SERGE VLĂDUŢ (U. Marseille)

THE MAIN MESSAGE OF THIS TUTORIAL:

- The task of node repair in distributed storage gives rise to a range of new, previously unrecognized problems in coding theory and related areas of computer science and discrete mathematics.
- These problems have been actively studied for the past decade and led to the emergence of new methods and ideas in these areas.
- The goal of this tutorial is to introduce these methods and the associated results as well as to point out new research directions.

ACKNOWLEDGMENTS:
ITZHAK TAMO (Tel Aviv U.)
MIN YE (Princeton U.)
CHEN ZITAN (UMD)
SERGE VLĂDUŢ (U. Marseille)



THE MAIN MESSAGE OF THIS TUTORIAL:

- The task of node repair in distributed storage gives rise to a range of new, previously unrecognized problems in coding theory and related areas of computer science and discrete mathematics.
- These problems have been actively studied for the past decade and led to the emergence of new methods and ideas in these areas.
- The goal of this tutorial is to introduce these methods and the associated results as well as to point out new research directions.

ACKNOWLEDGMENTS:

ITZHAK TAMO (Tel Aviv U.)

MIN YE (Princeton U.)

CHEN ZITAN (UMD)

SERGE VLĂDUŢ (U. Marseille)




A Data Center

A Data Center

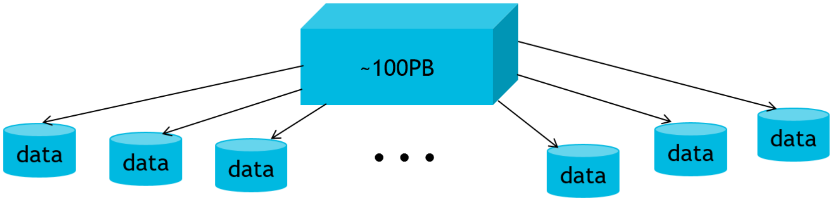
Data Center Equipment

- Power backup provided by 1.75 generator capacity being fed from 4000 gallon diesel fuel cells
- N+1 redundant HVAC system using multiple units with backup units standing by
  Redundant cable routing system
- Anti-static environment
- Triple power feeds, UPS
- N+1 cooling units
- N+1 UPS Systems
- Rated to withstand Class 3 – 4 hurricane strength
- NOC (Network Operations Center) staffed with senior system technicians 24 x 7 x 365

Network

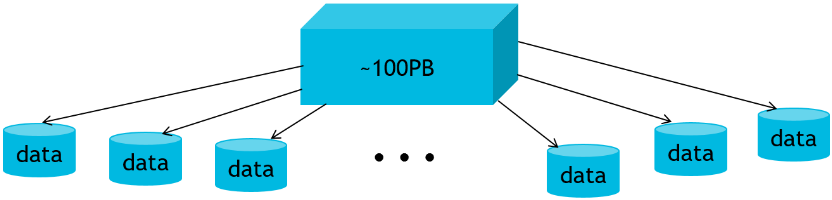
- 100 Gigabit Ethernet Core
- All switch based internal network
- 100% network uptime guarantee
- Current generation Terrathon class routers with terrabit routing capacity.
- Backbone connections from Level 3, MCI, and Time Warner
- Over 300 direct private peering relationships to ensure the best delivery of your traffic
- Over 77 Gbps in total bandwidth capacity
- Servers are placed in segmented network
- Up to10000 Mbps port, fully burstable

Motivation: Distributed Storage Systems (DSS)



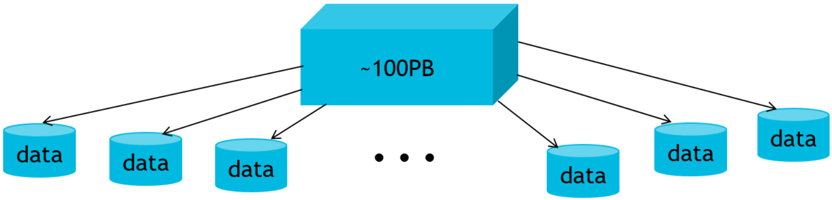
- DSS spread data across thousands of storage nodes

Motivation: Distributed Storage Systems (DSS)



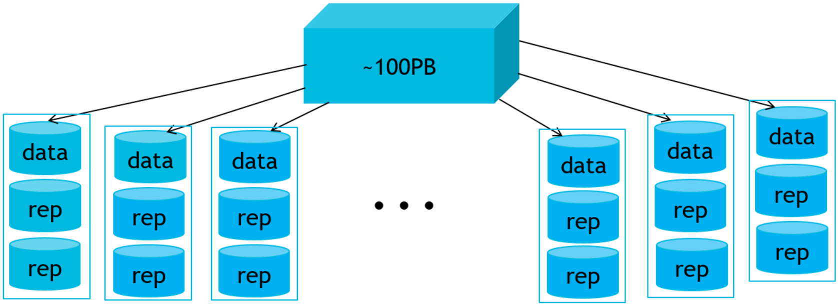
- DSS spread data across thousands of storage nodes
- Individual storage nodes fail frequently

Motivation: Distributed Storage Systems (DSS)



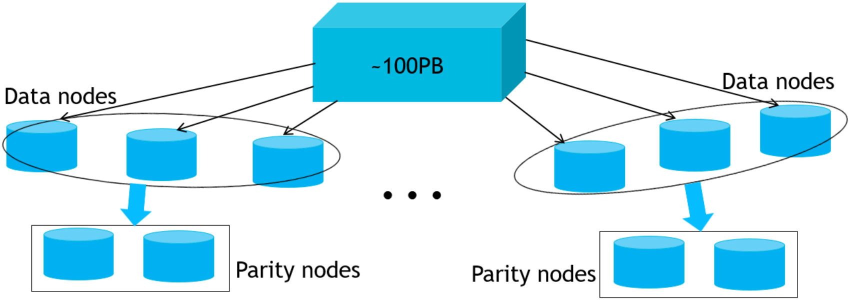
- DSS spread data across thousands of storage nodes
- Individual storage nodes fail frequently
- To protect the data we rely on erasure codes

Replication: large storage overhead



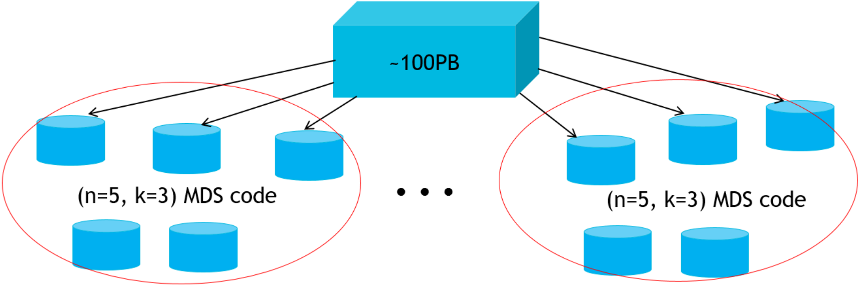
Can tolerate any 2 node failures
Storage overhead $= \quad 3\times$

MDS codes: Optimal storage efficiency



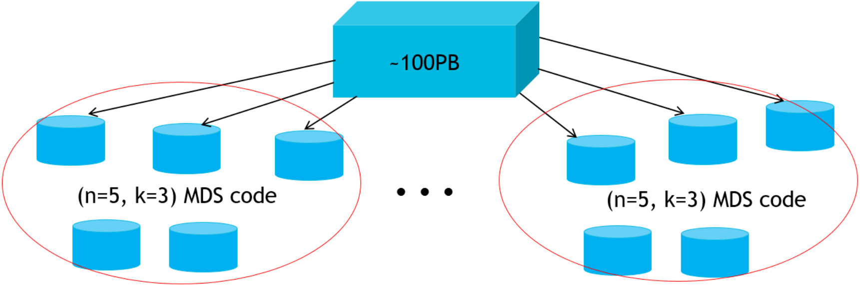
Add 2 parity nodes to every 3 data nodes
Form an $(n = 5, k = 3)$ MDS code

MDS codes: Optimal storage efficiency



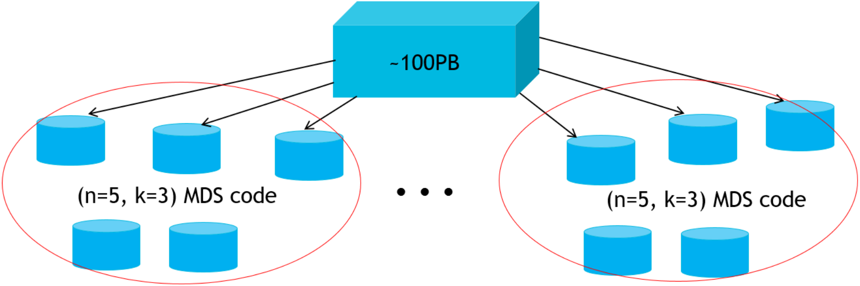
Can tolerate any 2 node failures
Storage overhead $= \quad 1.67\times$

MDS codes: Optimal storage efficiency



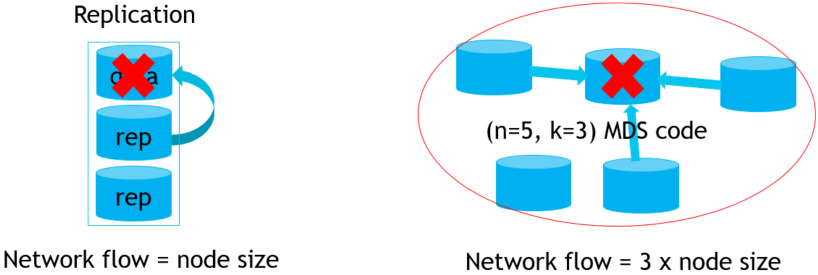
Block-based system model: Data blocks are encoded independently

MDS codes: Optimal storage efficiency



Each node stores l symbols of the block

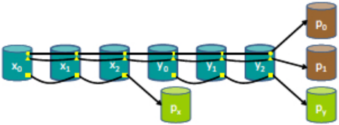
Repair Bandwidth



MDS code uses much more network bandwidth during data regeneration

Repair degree

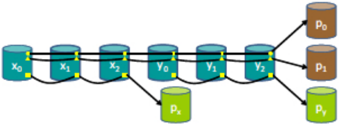
A combination of local and global parity checks for single and multiple nodes failures



(**C. Huang** at al., Erasure coding in Windows Azure Storage, USENIX Conf. 2012)

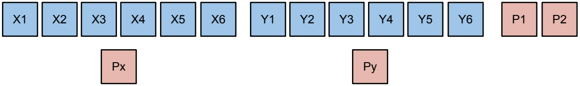
Repair degree

A combination of local and global parity checks for single and multiple nodes failures



(**C. Huang** at al., Erasure coding in Windows Azure Storage, USENIX Conf. 2012)

Other similar constructions (Windows Azure code)



Pyramid codes (**C. Huang** et al., 2007)

Main ideas

REPAIR DEGREE

- P. GOPALAN, C. HUANG, H. SIMITCI, AND S. YEKHANIN, *On the locality of codeword symbols*, T-IT, 2012
  - C. HUANG, M. CHEN, AND J. LI, Pyramid codes, Proc. 6th IEEE Int. Symp. NCA, 2007

REPAIR BANDWIDTH

- A.G. DIMAKIS, P.B. GODDFREY, Y. WU, M.J. WAINWRIGHT, AND K. RAMCHANDRAN, *Network coding for distributed storage*, T-IT, 2010

REPAIR OF RS CODES

- V. GURUSWAMI AND M. WOOTTERS, *Repairing Reed-Solomon codes*, T-IT, 2017

Different versions of the repair problem

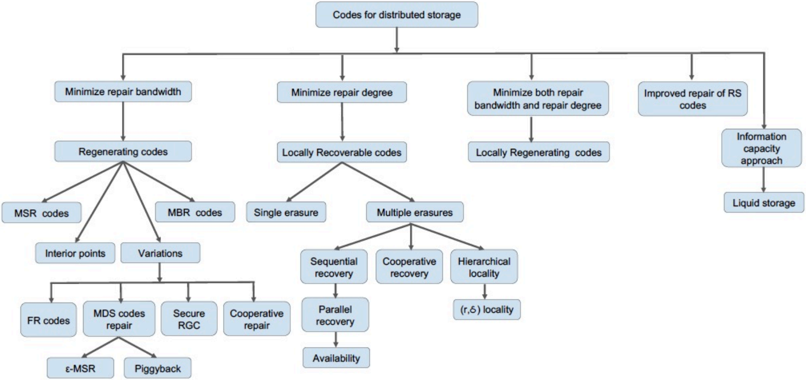


Figure 1 from "Erasure coding for distributed storage: An overview"

S.B. Balaji et al., arXiv:1806.04437

Different versions of the repair problem

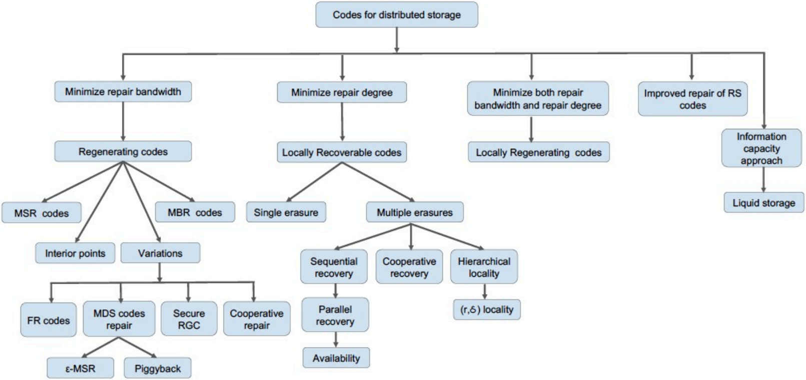


Figure 1 from "Erasure coding for distributed storage: An overview"

S.B. BALAJI et al., arXiv:1806.04437

Current literature count is in the hundreds

Outline of the tutorial

1. Repair degree
    1.1 Locally recoverable codes and their parameters
    1.2 Constructions for low repair degree
    1.3 Related problems:
        1.3.1 availability,
        1.3.2 repair of several nodes,
        1.3.3 hierarchical locality,
        1.3.4 sequential recovery
    1.4 Open questions: MR codes, maximum length of optimal LRC codes
2. Repair bandwidth
    2.1 Information flow graphs and the cutset bound
        2.1.1 Regenerating codes
        2.1.2 The MSR case
        2.1.3 Construction of MSR codes
        2.1.4 Multiple erasures: Centralized and cooperative repair
        2.1.5 Optimal access and subpacketization
        2.1.6 Repair of Reed-Solomon codes
    2.2 Heterogeneous storage
3. Looking forward: Storage networks, random failures

Correcting one erasure

Correcting one erasure

- A linear code $\mathcal{C}$ corrects one erasure if and only if it does not contain codewords of Hamming weight 1.

Correcting one erasure

- A linear code $\mathcal{C}$ corrects one erasure if and only if it does not contain codewords of Hamming weight 1.
- Consider a binary code that encodes 3-bit messages into 7-bit code blocks:

$$(110) \times \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$
$$(0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0)$$

Correcting one erasure

- A linear code $\mathcal{C}$ corrects one erasure if and only if it does not contain codewords of Hamming weight 1.
- Consider a binary code that encodes 3-bit messages into 7-bit code blocks:

$$(110) \times \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$(0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0)$$

> Any 4 bits identify the codeword uniquely (in other words, the code can correct up to 3 erasures)

Correcting one erasure

- A linear code $\mathcal{C}$ corrects one erasure if and only if it does not contain codewords of Hamming weight 1.
- Consider a binary code that encodes 3-bit messages into 7-bit code blocks:

$$(110) \times \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$
$$(0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0)$$

Any 4 bits identify the codeword uniquely (in other words, the code can correct up to 3 erasures)

- At the same time, a single erasure can be recovered from 2 bits: For instance

$$(0 \quad \cancel{1} \quad 1 \quad 1 \quad 1 \quad 0 \quad 0)$$

$$c_2 = c_4 \oplus c_6$$

Correcting one erasure

Correcting one erasure

- We are interested in efficient implementation of erasure correction motivated by applications in storage

Correcting one erasure

- We are interested in efficient implementation of erasure correction motivated by applications in storage
- Coordinates of the codeword $C = (C_1, \dots, C_n) \in \mathcal{C}$ are called nodes; If a node C_i is erased (failed), we look at the other (surviving) coordinates, called helper nodes, and download their values, or functions of these values.

Reminder: Finite fields

Reminder: Finite fields

- Consider the set of integers modulo 13

$$\mathbb{F}_{13} = \{0, 1, 2, \dots, 12\}$$

with operations $a + b \,(\text{mod} 13), a \cdot b \,(\text{mod} 13)$

Reminder: Finite fields

- Consider the set of integers modulo 13

$$\mathbb{F}_{13} = \{0, 1, 2, \dots, 12\}$$

  with operations $a + b \,(\mathrm{mod} 13), a \cdot b \,(\mathrm{mod} 13)$
- It is possible to subtract and divide:
  if $6a = 2$ then $a = 6^{-1} \cdot 2 = 11 \cdot 2 = 9$

Reminder: Finite fields

- Consider the set of integers modulo 13

$$\mathbb{F}_{13} = \{0, 1, 2, \dots, 12\}$$

  with operations $a + b \,(\mathrm{mod} 13), a \cdot b \,(\mathrm{mod} 13)$
- It is possible to subtract and divide:
  if $6a = 2$ then $a = 6^{-1} \cdot 2 = 11 \cdot 2 = 9$
- A field is a set of elements that are closed under addition and multiplication, and support the inverse operations

Reminder: Finite fields

- Consider the set of integers modulo 13

$$\mathbb{F}_{13} = \{0, 1, 2, \ldots, 12\}$$

  with operations $a + b \,(\text{mod} 13), a \cdot b \,(\text{mod} 13)$
- It is possible to subtract and divide:
  if $6a = 2$ then $a = 6^{-1} \cdot 2 = 11 \cdot 2 = 9$
- A field is a set of elements that are closed under addition and multiplication, and support the inverse operations
- Finite fields exist for any $q = p^m$, where p is a prime and $m \geqslant 1$

Reminder: Finite fields

- Consider the set of integers modulo 13

$$\mathbb{F}_{13} = \{0, 1, 2, \ldots, 12\}$$

  with operations $a + b \,(\text{mod} 13), a \cdot b \,(\text{mod} 13)$

- It is possible to subtract and divide:
  if $6a = 2$ then $a = 6^{-1} \cdot 2 = 11 \cdot 2 = 9$
- A field is a set of elements that are closed under addition and multiplication, and support the inverse operations
- Finite fields exist for any $q = p^m$, where p is a prime and $m \geqslant 1$
- For any given q the field $\mathbb{F}_q$ is unique

Reminder: Finite fields

- Consider the set of integers modulo 13

$$\mathbb{F}_{13} = \{0, 1, 2, \dots, 12\}$$

  with operations $a + b\,(\mathrm{mod} 13), a \cdot b\,(\mathrm{mod} 13)$

- It is possible to subtract and divide:
  if $6a = 2$ then $a = 6^{-1} \cdot 2 = 11 \cdot 2 = 9$
- A field is a set of elements that are closed under addition and multiplication, and support the inverse operations
- Finite fields exist for any $q = p^m$, where p is a prime and $m \geqslant 1$
- For any given q the field $\mathbb{F}_q$ is unique
- Polynomials and linear algebra work over $\mathbb{F}_q$ in many ways as over $\mathbb{R}$

Reminder: Reed-Solomon codes

Reminder: Reed-Solomon codes

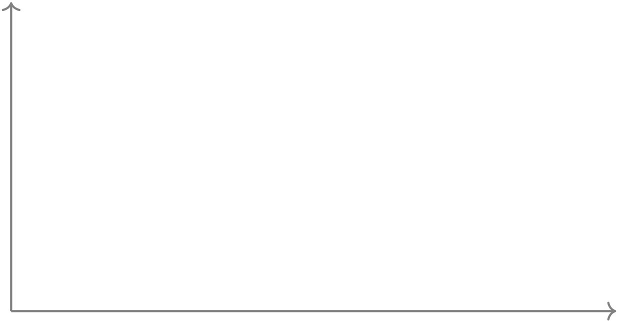
(a_0, a_1, a_2, a_3)

Reminder: Reed-Solomon codes

$$(a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \rightarrow f(x) = (f(P_1), f(P_2), ..., f(P_n))$$

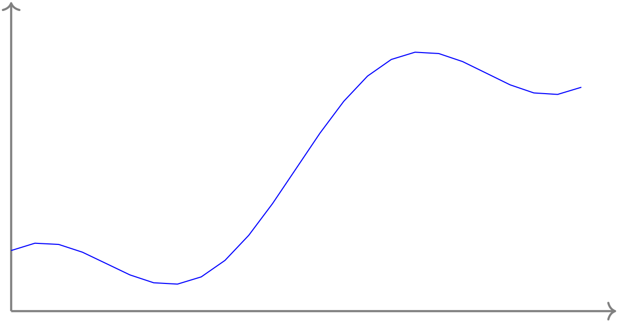
Reminder: Reed-Solomon codes

$$(a_0, a_1, a_2, a_3) \to f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \to f(x) = (f(P_1), f(P_2), \ldots, f(P_n))$$



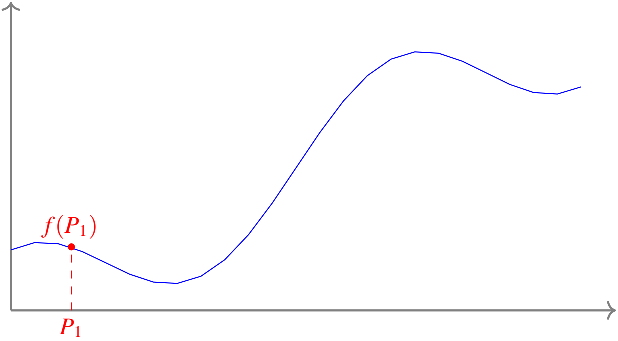
Reminder: Reed-Solomon codes

$$(a_0, a_1, a_2, a_3) \to f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \to f(x) = (f(P_1), f(P_2), \ldots, f(P_n))$$



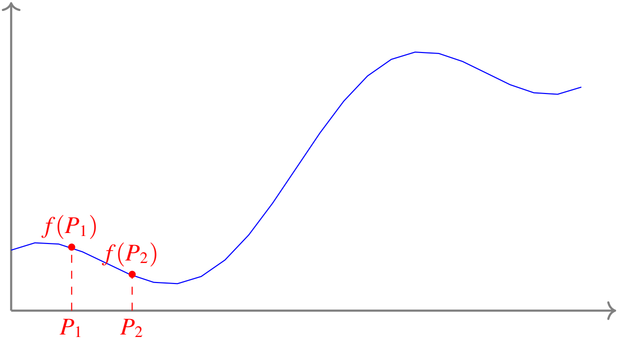
Reminder: Reed-Solomon codes

$$(a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \rightarrow f(x) = (f(P_1), f(P_2), \ldots, f(P_n))$$



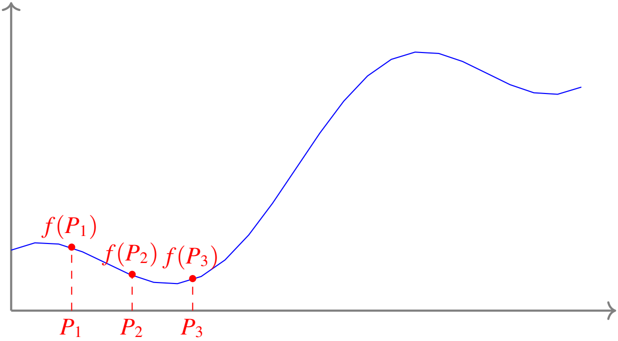
Reminder: Reed-Solomon codes

$$(a_0, a_1, a_2, a_3) \to f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \to f(x) = (f(P_1), f(P_2), ..., f(P_n))$$



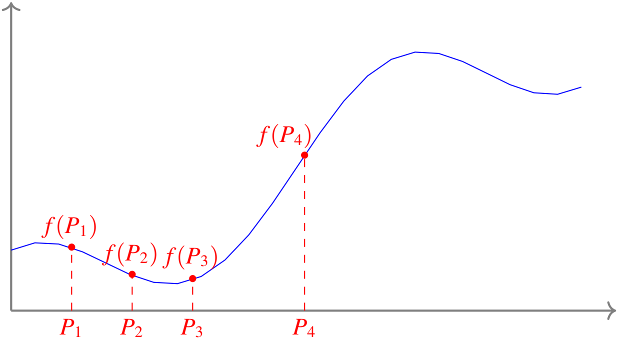
Reminder: Reed-Solomon codes

$$(a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \rightarrow f(x) = (f(P_1), f(P_2), ..., f(P_n))$$



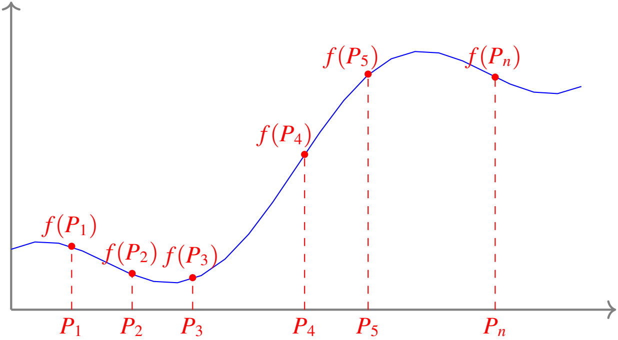
Reminder: Reed-Solomon codes

$$(a_0, a_1, a_2, a_3) \to f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \to f(x) = (f(P_1), f(P_2), \ldots, f(P_n))$$



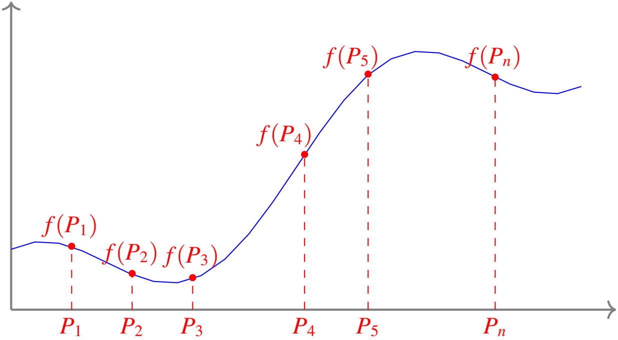
Reminder: Reed-Solomon codes

$$(a_0, a_1, a_2, a_3) \to f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \to f(x) = (f(P_1), f(P_2), \ldots, f(P_n))$$



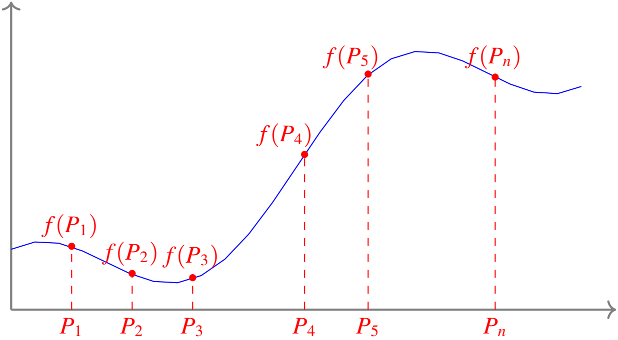
Reminder: Reed-Solomon codes

$$(a_0, a_1, a_2, a_3) \to f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \to f(x) = (f(P_1), f(P_2), \ldots, f(P_n))$$



Reminder: Reed-Solomon codes

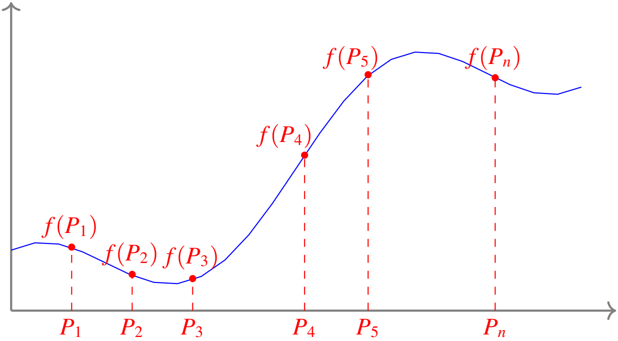
$$(a_0, a_1, a_2, a_3) \to f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \to f(x) = (f(P_1), f(P_2), \ldots, f(P_n))$$



At most 3 values $f(P_i)$ can be 0; thus the Hamming weight of the codeword eval$(f(x))$ is $n-3$.

Reminder: Reed-Solomon codes

$$(a_0, a_1, a_2, a_3) \to f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \to f(x) = (f(P_1), f(P_2), \ldots, f(P_n))$$



At most 3 values $f(P_i)$ can be 0; thus the Hamming weight of the codeword eval$(f(x))$ is $n-3$.

An RS code is a set of vectors obtained by evaluating all polynomials of degree up to $k-1$.
The ***minimum distance*** of the RS code is $n-(k-1)$; and this is the largest possible value according to the Singleton bound (MDS code).

Reed-Solomon codes

An RS code $\mathcal{C}$ is a linear code of length $n \leqslant q-1$ over the field $\mathbb{F}_q$

Reed-Solomon codes

An RS code $\mathcal{C}$ is a linear code of length $n \leqslant q-1$ over the field $\mathbb{F}_q$

Given a polynomial $f \in \mathbb{F}_q[x]$ and a set $A = \{P_1, \dots, P_n\} \subset \mathbb{F}_q$ define the map

$$ev_A : f \mapsto (f(P_i), i = 1, \dots, n)$$

Reed-Solomon codes

An RS code $\mathcal{C}$ is a linear code of length $n \leqslant q-1$ over the field $\mathbb{F}_q$

Given a polynomial $f \in \mathbb{F}_q[x]$ and a set $A = \{P_1, \dots, P_n\} \subset \mathbb{F}_q$ define the map

$$ev_A : f \mapsto (f(P_i), i = 1, \dots, n)$$

RS code $\mathcal{C}$ encodes messages of k symbols.

Let $V_k(q) = \{f \in \mathbb{F}_q[x] : \deg(f) \leqslant k-1\}$

$$\begin{aligned} \mathcal{C} : V_k(q) &\to \mathbb{F}_q^n \\ f &\mapsto ev_A(f) = (f(P_i), i = 1, \dots, n) \end{aligned}$$

Reed-Solomon codes

An RS code $\mathcal{C}$ is a linear code of length $n \leqslant q-1$ over the field $\mathbb{F}_q$

Given a polynomial $f \in \mathbb{F}_q[x]$ and a set $A = \{P_1, \dots, P_n\} \subset \mathbb{F}_q$ define the map

$$ev_A : f \mapsto (f(P_i), i = 1, \dots, n)$$

RS code $\mathcal{C}$ encodes messages of k symbols.

Let $V_k(q) = \{f \in \mathbb{F}_q[x] : \deg(f) \leqslant k-1\}$

$$\begin{aligned} \mathcal{C} : V_k(q) &\to \mathbb{F}_q^n \\ f &\mapsto ev_A(f) = (f(P_i), i = 1, \dots, n) \end{aligned}$$

Example: Let $q = 8, f(x) = 1 + \alpha x + \alpha x^2$

$$f(x) \mapsto (1, \alpha^4, \alpha^6, \alpha^4, \alpha, \alpha, \alpha^6)$$

Limitations of Reed-Solomon codes

Limitations of Reed-Solomon codes

Example: $[14, 10]$ RS code

Limitations of Reed-Solomon codes

Example: $[14, 10]$ RS code

Limitations of Reed-Solomon codes

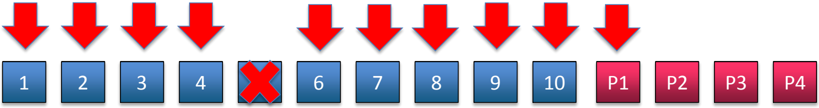
Example: $[14, 10]$ RS code

- Loss of a node triggers the repair task

Limitations of Reed-Solomon codes

Example: $[14, 10]$ RS code

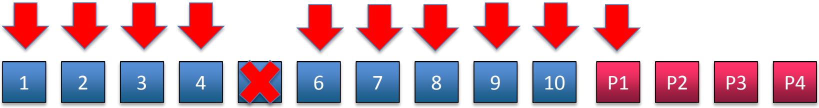
- Loss of a node triggers the repair task
- Need to transmit information from 10 nodes to recover one lost node



Limitations of Reed-Solomon codes

Example: $[14, 10]$ RS code

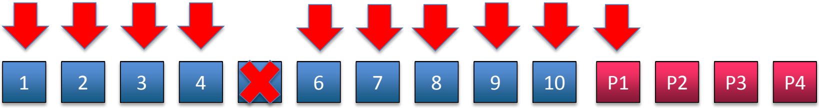
- Loss of a node triggers the repair task
- Need to transmit information from 10 nodes to recover one lost node
- Generates 10x more traffic compared to replication for recovery of one node



Limitations of Reed-Solomon codes

Example: $[14, 10]$ RS code

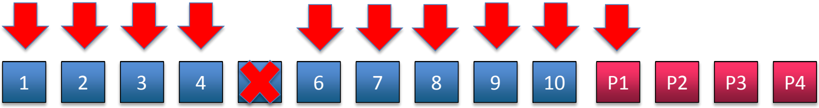
- Loss of a node triggers the repair task
- Need to transmit information from 10 nodes to recover one lost node
- Generates 10x more traffic compared to replication for recovery of one node
- If a large portion of the data is RS-coded $\Longrightarrow$ saturation of the network



Limitations of Reed-Solomon codes

Example: $[14, 10]$ RS code

- Loss of a node triggers the repair task
- Need to transmit information from 10 nodes to recover one lost node
- Generates 10x more traffic compared to replication for recovery of one node
- If a large portion of the data is RS-coded $\Longrightarrow$ saturation of the network
- **Goal:** Construct efficient codes with "good" repair process



Locally Recoverable Codes - Definition

(n, k, r) **LRC Code**

Locally Recoverable Codes - Definition

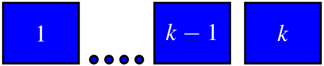
(n, k, r) **LRC Code**

- Takes k blocks (symbols) $\rightarrow$ produces n blocks

Locally Recoverable Codes - Definition

(n, k, r) **LRC Code**

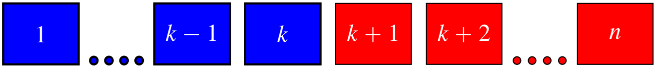
- Takes k blocks (symbols) $\rightarrow$ produces n blocks



Locally Recoverable Codes - Definition

(n, k, r) **LRC Code**

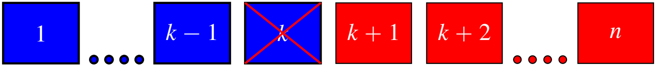
- Takes k blocks (symbols) $\rightarrow$ produces n blocks



Locally Recoverable Codes - Definition

(n, k, r) **LRC Code**

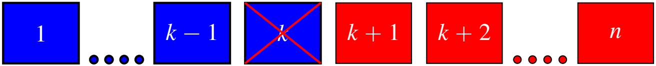
- Takes k blocks (symbols) $\rightarrow$ produces n blocks
- An erasure has occurred



Locally Recoverable Codes - Definition

(n, k, r) **LRC Code**

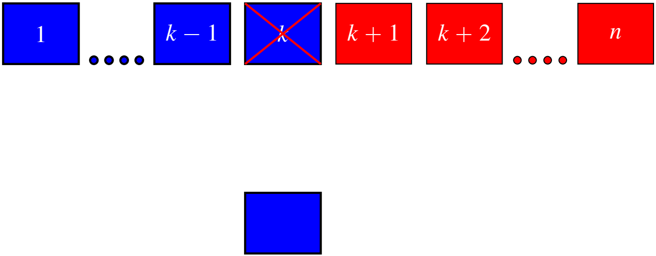
- Takes k blocks (symbols) $\rightarrow$ produces n blocks
- An erasure has occurred
- Every symbol i has a recovering set $\mathcal{R}_i$ of r other symbols, $r \ll k$



Locally Recoverable Codes - Definition

(n, k, r) **LRC Code**

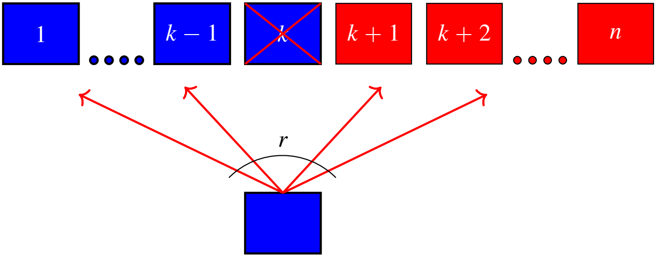
- Takes k blocks (symbols) $\rightarrow$ produces n blocks
- An erasure has occurred
- Every symbol i has a recovering set $\mathcal{R}_i$ of r other symbols, $r \ll k$



Locally Recoverable Codes - Definition

(n, k, r) **LRC Code**

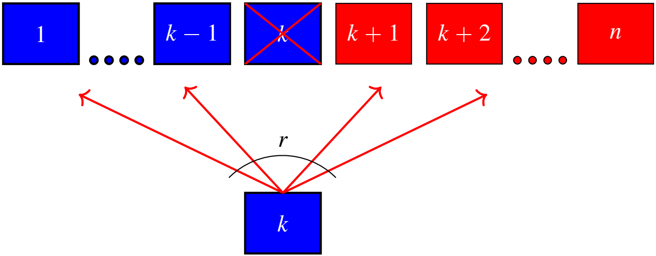
- Takes k blocks (symbols) $\rightarrow$ produces n blocks
- An erasure has occurred
- Every symbol i has a recovering set $\mathcal{R}_i$ of r other symbols, $r \ll k$



Locally Recoverable Codes - Definition

(n, k, r) **LRC Code**

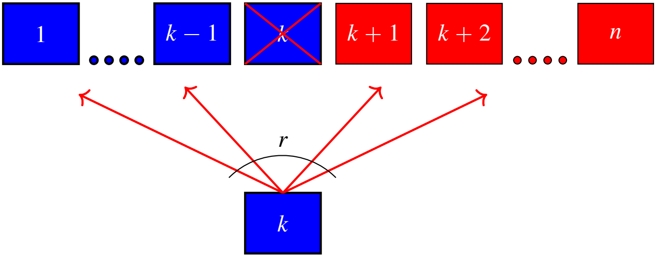
- Takes k blocks (symbols) $\rightarrow$ produces n blocks
- An erasure has occurred
- Every symbol i has a recovering set $\mathcal{R}_i$ of r other symbols, $r \ll k$



Locally Recoverable Codes - Definition

(n, k, r) **LRC Code**

- Takes k blocks (symbols) $\rightarrow$ produces n blocks
- An erasure has occurred
- Every symbol i has a recovering set $\mathcal{R}_i$ of r other symbols, $r \ll k$
- Clearly $1 \leqslant r \leqslant k$



Parameters of LRC codes

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$

Parameters of LRC codes

Let $\mathcal{C}$ be an (n,k,r) LRC code

- Assume $r|k$ and $(r+1)|n$
- Rate?

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- Rate?
- Minimum distance?

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The rate is bounded by

$$\frac{k}{n} \leqslant \frac{r}{r+1}.$$

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The rate is bounded by
$$\frac{k}{n} \leqslant \frac{r}{r+1}.$$

Proof:

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The rate is bounded by

$$\frac{k}{n} \leqslant \frac{r}{r+1}.$$

Proof:

- There exist at most $\frac{nr}{r+1}$ coordinates that determine the exact codeword

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$

- The rate is bounded by
$$\frac{k}{n} \leqslant \frac{r}{r+1}.$$

Proof:

- There exist at most $\frac{nr}{r+1}$ coordinates that determine the exact codeword

- This follows since iteratively:

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The rate is bounded by

$$\frac{k}{n} \leqslant \frac{r}{r+1}.$$

Proof:

- There exist at most $\frac{nr}{r+1}$ coordinates that determine the exact codeword
- This follows since iteratively:

  1. Cost: expose the values of the coordinates in a recovering set $\mathcal{R}_i$, $|\mathcal{R}_i| \leqslant r$

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The rate is bounded by

$$\frac{k}{n} \leqslant \frac{r}{r+1}.$$

Proof:

- There exist at most $\frac{nr}{r+1}$ coordinates that determine the exact codeword
- This follows since iteratively:
  1. Cost: expose the values of the coordinates in a recovering set $\mathcal{R}_i$, $|\mathcal{R}_i| \leqslant r$
  2. Free: the value of the i-th coordinate

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The rate is bounded by

$$\frac{k}{n} \leqslant \frac{r}{r+1}.$$

Proof:

- There exist at most $\frac{nr}{r+1}$ coordinates that determine the exact codeword
- This follows since iteratively:

1. Cost: expose the values of the coordinates in a recovering set $\mathcal{R}_i$, $|\mathcal{R}_i| \leqslant r$
2. Free: the value of the i-th coordinate
3. Upon exposing at most $\frac{nr}{r+1}$ coordinates, we recover the entire codeword

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The rate is bounded by

$$\frac{k}{n} \leqslant \frac{r}{r+1}.$$

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The rate is bounded by

$$\frac{k}{n} \leqslant \frac{r}{r+1}.$$

- The bound is tight (even over $\mathbb{F}_2$)

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The rate is bounded by
$$\frac{k}{n} \leqslant \frac{r}{r+1}.$$
- The bound is tight (even over $\mathbb{F}_2$)
  - Partition the k bits into k/r sets of size r

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The rate is bounded by
$$\frac{k}{n} \leqslant \frac{r}{r+1}.$$
- The bound is tight (even over $\mathbb{F}_2$)
  - Partition the k bits into k/r sets of size r
  - Add parity check bit to each set

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The minimum distance is bounded by

$$d \leqslant n - k - \left\lceil \frac{k}{r} \right\rceil + 2$$

P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, T-IT 2012
D. Papailiopoulos and A. Dimakis, ISIT 2012

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The minimum distance is bounded by

$$d \leqslant n - k - \left\lceil \frac{k}{r} \right\rceil + 2$$

P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, T-IT 2012
D. Papailiopoulos and A. Dimakis, ISIT 2012

Remarks:

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The minimum distance is bounded by

$$d \leqslant n - k - \left\lceil \frac{k}{r} \right\rceil + 2$$

P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, T-IT 2012
D. Papailiopoulos and A. Dimakis, ISIT 2012

Remarks:

- Smaller locality $\implies$ lower failure resilience

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The minimum distance is bounded by

$$d \leqslant n - k - \left\lceil \frac{k}{r} \right\rceil + 2$$

P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, T-IT 2012
D. Papailiopoulos and A. Dimakis, ISIT 2012

Remarks:

- Smaller locality $\Longrightarrow$ lower failure resilience
- Generalization of the *Singleton* bound $(r = k)$

Parameters of LRC codes

Let $\mathcal{C}$ be an (n, k, r) LRC code

- Assume $r|k$ and $(r+1)|n$
- The minimum distance is bounded by

$$d \leqslant n - k - \left\lceil \frac{k}{r} \right\rceil + 2$$

P. GOPALAN, C. HUANG, H. SIMITCI, AND S. YEKHANIN, T-IT 2012
D. PAPAILIOPOULOS AND A. DIMAKIS, ISIT 2012

Remarks:

- Smaller locality $\implies$ lower failure resilience
- Generalization of the *Singleton* bound $(r = k)$
- Optimal (n, k, r) LRC code achieves the bound with equality

The distance bound

Main idea.

Let $\mathcal{C}$ be a q-ary code of length n, size q^k. The distance $d(\mathcal{C})$ satisfies

$$d(\mathcal{C}) \leqslant n - \{|S| : |\mathcal{C}_S| < q^k\}$$

Details:

- $\frac{k}{n} \leqslant \frac{r}{r+1} \implies \exists$ a set I of $\lfloor \frac{k-1}{n} \rfloor$ redundant coordinates
- Set $\mathcal{R} = \cup_{i\in I}\mathcal{R}_i$. Clearly $|\mathcal{R}| \leqslant k-1$
- If $|\mathcal{R}| < k-1$ add to it

The Singleton bound (with locality):

Let $I_i \subset [n], |I_i| \leqslant r$ be the recovery set for the symbol $c_i, i = 1, \dots, n$.

Let $J_m = \cup_{i=1}^m I_i$, where $m = \lfloor (k-1)/r \rfloor$. Clearly $|J_m| \leqslant k-1$.

Consider the subset $J_m' = J_m \cup \{1, \dots, m\}$. We have $\mathcal{C}_{J_m'} \leqslant q^{k-1}$.

If $|J_m'| < k-1$, add to J_m' any $k-1-|J_m|$ other coordinates to form the set $L_m \subset [n]$.

We have

$$|\mathcal{C}_{L_m}| < q^k$$
$$|L_m| = k-1+m = k-1+\left\lfloor \frac{k-1}{r} \right\rfloor = k-2+\left\lceil \frac{k}{r} \right\rceil$$

Constructing optimal LRC codes: Early results

Constructing optimal LRC codes: Early results

- Early constructions:

Constructing optimal LRC codes: Early results

- Early constructions:

  1. Non explicit

Constructing optimal LRC codes: Early results

- Early constructions:

  1. Non explicit

  2. Field size is superpolynomial in the length

Constructing optimal LRC codes: Early results

- Early constructions:
  1. Non explicit
  2. Field size is superpolynomial in the length
- Optimal $((r+1)\lceil \frac{k}{r} \rceil, k, r)$ LRC code [PRASANTH, KAMATH, LALITHA, AND KUMAR,2012]

Constructing optimal LRC codes: Early results

- Early constructions:

    1. Non explicit

    2. Field size is superpolynomial in the length

- Optimal $((r+1)\lceil\frac{k}{r}\rceil, k, r)$ LRC code [Prasanth, Kamath, Lalitha, and Kumar,2012]

- Explicit constructions [Rawat, Koyluoglu, Silberstein, Vishwanath 2014, Gopalan, Huang, Jenkins, Yekhanin 2014, Tamo, Papailiopoulos, Dimakis 2014]

Constructing optimal LRC codes: Early results

- Early constructions:
  1. Non explicit
  2. Field size is superpolynomial in the length
- Optimal $((r+1)\lceil \frac{k}{r} \rceil, k, r)$ LRC code [PRASANTH, KAMATH, LALITHA, AND KUMAR,2012]
- Explicit constructions [RAWAT, KOYLUOGLU, SILBERSTEIN, VISHWANATH 2014, GOPALAN, HUANG, JENKINS, YEKHANIN 2014, TAMO, PAPAILIOPOULOS, DIMAKIS 2014]
  1. Any n, k, r

Constructing optimal LRC codes: Early results

- Early constructions:
  1. Non explicit
  2. Field size is superpolynomial in the length
- Optimal $((r+1)\lceil\frac{k}{r}\rceil, k, r)$ LRC code [Prasanth, Kamath, Lalitha, and Kumar,2012]
- Explicit constructions [Rawat, Koyluoglu, Silberstein, Vishwanath 2014, Gopalan, Huang, Jenkins, Yekhanin 2014, Tamo, Papailiopoulos, Dimakis 2014]
  1. Any n, k, r
  2. Field size is superpolynomial

Optimal LRC codes - Easy cases

Optimal LRC codes - Easy cases

- $r = k$

Optimal LRC codes - Easy cases

- $r = k$

  1. $d \leqslant n - k + 1$

Optimal LRC codes - Easy cases

- $r = k$

  1. $d \leqslant n - k + 1$
  2. An (n, k) RS is an (n, k, k) optimal LRC code

Optimal LRC codes - Easy cases

- $r = k$

    1. $d \leqslant n - k + 1$
    2. An (n, k) RS is an (n, k, k) optimal LRC code
    3. $|\mathbb{F}| = O(n)$

Optimal LRC codes - Easy cases

- $r = k$

  1. $d \leqslant n - k + 1$
  2. An (n, k) RS is an (n, k, k) optimal LRC code
  3. $|\mathbb{F}| = O(n)$

- $r = 1$

Optimal LRC codes - Easy cases

- $r = k$

  1. $d \leqslant n - k + 1$
  2. An (n, k) RS is an (n, k, k) optimal LRC code
  3. $|\mathbb{F}| = O(n)$

- $r = 1$

  1. $d \leqslant 2(\frac{n}{2} - k + 1)$

Optimal LRC codes - Easy cases

- $r = k$
  1. $d \leqslant n - k + 1$
  2. An (n, k) RS is an (n, k, k) optimal LRC code
  3. $|\mathbb{F}| = O(n)$

- $r = 1$
  1. $d \leqslant 2(\frac{n}{2} - k + 1)$
  2. Duplication of an $(n/2, k)$ RS is an $(n, k, 1)$ optimal LRC code

Optimal LRC codes - Easy cases

- $r = k$

  1. $d \leqslant n - k + 1$
  2. An (n, k) RS is an (n, k, k) optimal LRC code
  3. $|\mathbb{F}| = O(n)$

- $r = 1$

  1. $d \leqslant 2(\frac{n}{2} - k + 1)$
  2. Duplication of an $(n/2, k)$ RS is an $(n, k, 1)$ optimal LRC code
  3. $|\mathbb{F}| = O(n)$

Optimal LRC codes - Easy cases

- $r = k$
  1. $d \leqslant n - k + 1$
  2. An (n, k) RS is an (n, k, k) optimal LRC code
  3. $|\mathbb{F}| = O(n)$

- $r = 1$
  1. $d \leqslant 2(\frac{n}{2} - k + 1)$
  2. Duplication of an $(n/2, k)$ RS is an $(n, k, 1)$ optimal LRC code
  3. $|\mathbb{F}| = O(n)$

- Q: What happens for $1 < r < k$?

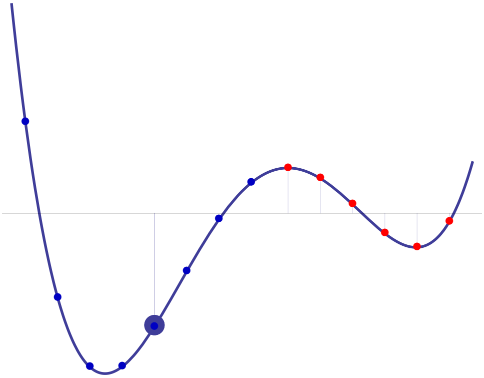
Optimal LRC codes - Easy cases

- $r = k$
  1. $d \leqslant n - k + 1$
  2. An (n, k) RS is an (n, k, k) optimal LRC code
  3. $|\mathbb{F}| = O(n)$

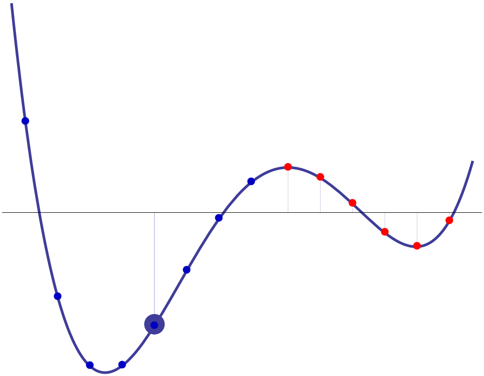
- $r = 1$
  1. $d \leqslant 2(\frac{n}{2} - k + 1)$
  2. Duplication of an $(n/2, k)$ RS is an $(n, k, 1)$ optimal LRC code
  3. $|\mathbb{F}| = O(n)$

- Q: What happens for $1 < r < k$?
- Q: Generalize the optimal codes for $r = 1, k$ to codes with arbitrary r?

Reed-Solomon codes



Reed-Solomon codes



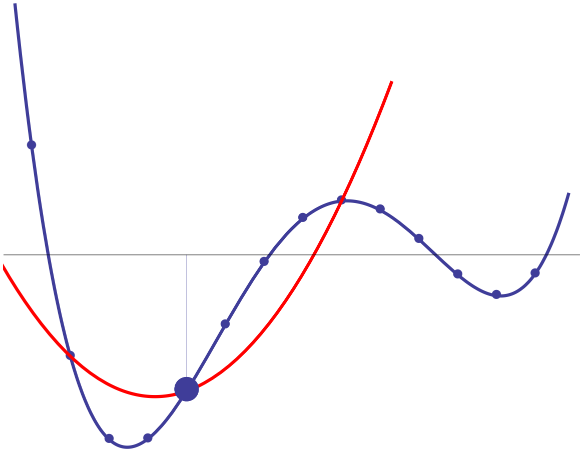
To recover one erased value we need to read k other values

LRC codes: Idea of construction

What if we can interpolate low-degree polynomials?

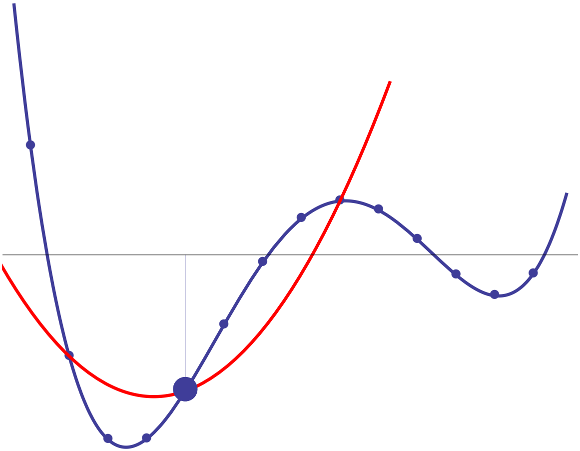
LRC codes: Idea of construction

What if we can interpolate low-degree polynomials?



LRC codes: Idea of construction

What if we can interpolate low-degree polynomials?



It is possible to construct such codes by carefully choosing subcodes of the RS codes

Construction of (n, k, r) LRC codes: Example

Construction of (n,k,r) LRC codes: Example

Parameters: $n = 9, k = 4, r = 2, q = 13;$

Construction of (n,k,r) LRC codes: Example

Parameters: $n = 9, k = 4, r = 2, q = 13;$

Set of points: A={1,2,3,4,5,6,9,10,12}

$$\mathcal{A} = \{A_1 = \{1,3,9\}, A_2 = \{2,6,5\}, A_3 = \{4,12,10\}\}$$

Construction of (n, k, r) LRC codes: Example

Parameters: $n = 9, k = 4, r = 2, q = 13;$

Set of points: A={1,2,3,4,5,6,9,10,12}

$$\mathcal{A} = \{A_1 = \{1, 3, 9\}, A_2 = \{2, 6, 5\}, A_3 = \{4, 12, 10\}\}$$

Message: $a = (a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) \in \mathbb{F}_q^k$

Construction of (n, k, r) LRC codes: Example

Parameters: $n = 9, k = 4, r = 2, q = 13;$

Set of points: A={1,2,3,4,5,6,9,10,12}

$$\mathcal{A} = \{A_1 = \{1, 3, 9\}, A_2 = \{2, 6, 5\}, A_3 = \{4, 12, 10\}\}$$

Message: $a = (a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) \in \mathbb{F}_q^k$

Polynomial space:

$$V_k(q) := \{a_{0,0} + a_{1,0}x + a_{0,1}x^3 + a_{1,1}x^4\}$$

Construction of (n, k, r) LRC codes: Example

Parameters: $n = 9, k = 4, r = 2, q = 13$;

Set of points: A={1,2,3,4,5,6,9,10,12}

$$\mathcal{A} = \{A_1 = \{1, 3, 9\}, A_2 = \{2, 6, 5\}, A_3 = \{4, 12, 10\}\}$$

Message: $a = (a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) \in \mathbb{F}_q^k$

Polynomial space:

$$V_k(q) := \{a_{0,0} + a_{1,0}x + a_{0,1}x^3 + a_{1,1}x^4\}$$

E.g., $a = (1, 1, 1, 1), f_a(x) = 1 + x + x^3 + x^4; ev_A(f) = (4, 8, 7, 1, 11, 2, 0, 0, 0)$

Construction of (n, k, r) LRC codes: Example

Parameters: $n = 9, k = 4, r = 2, q = 13;$

Set of points: A={1,2,3,4,5,6,9,10,12}

$$\mathcal{A} = \{A_1 = \{1, 3, 9\}, A_2 = \{2, 6, 5\}, A_3 = \{4, 12, 10\}\}$$

Message: $a = (a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) \in \mathbb{F}_q^k$

Polynomial space:

$$V_k(q) := \{a_{0,0} + a_{1,0}x + a_{0,1}x^3 + a_{1,1}x^4\}$$

E.g., $a = (1, 1, 1, 1), f_a(x) = 1 + x + x^3 + x^4; ev_A(f) = (4, 8, 7, 1, 11, 2, 0, 0, 0)$

Say $c_1 = f_a(1)$ is erased. We access the recovering set A_1 to construct a line $\delta(x) = 2x + 2$ such that $\delta(3) = 8, \delta(9) = 7$.

Construction of (n, k, r) LRC codes: Example

Parameters: $n = 9, k = 4, r = 2, q = 13;$

Set of points: A={1,2,3,4,5,6,9,10,12}

$$\mathcal{A} = \{A_1 = \{1, 3, 9\}, A_2 = \{2, 6, 5\}, A_3 = \{4, 12, 10\}\}$$

Message: $a = (a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) \in \mathbb{F}_q^k$

Polynomial space:

$$V_k(q) := \{a_{0,0} + a_{1,0}x + a_{0,1}x^3 + a_{1,1}x^4\}$$

E.g., $a = (1, 1, 1, 1), f_a(x) = 1 + x + x^3 + x^4; ev_A(f) = (4, 8, 7, 1, 11, 2, 0, 0, 0)$

Say $c_1 = f_a(1)$ is erased. We access the recovering set A_1 to construct a line $\delta(x) = 2x + 2$ such that $\delta(3) = 8, \delta(9) = 7$.

Compute c_1 as $\delta(1) = 4$

Construction of (n, k, r) LRC codes

Assume that $q \geqslant n, (r+1)|n, r|k$

Let $A \subseteq \mathbb{F}_q, |A| = n$

Construction of (n, k, r) LRC codes

Assume that $q \geqslant n, (r+1)|n, r|k$
Let $A \subseteq \mathbb{F}_q, |A| = n$

Suppose there exists a polynomial $g(x) \in \mathbb{F}[x]$ such that

1. $\deg g = r + 1,$
2. There exists a partition $\mathcal{A} = \{A_1, ..., A_{\frac{n}{r+1}}\}$ of A into sets of size $r+1$, such that g is constant on each set A_i in the partition. For all $i = 1, \dots, n/(r+1)$, and any $\alpha, \beta \in A_i$,

$$g(\alpha) = g(\beta).$$

Construction of (n, k, r) LRC codes

Assume that $q \geqslant n, (r+1)|n, r|k$
Let $A \subseteq \mathbb{F}_q, |A| = n$

Suppose there exists a polynomial $g(x) \in \mathbb{F}[x]$ such that

1. $\deg g = r + 1$,
2. There exists a partition $\mathcal{A} = \{A_1, ..., A_{\frac{n}{r+1}}\}$ of A into sets of size $r+1$, such that g is constant on each set A_i in the partition. For all $i = 1, \dots, n/(r+1)$, and any $\alpha, \beta \in A_i$,

$$g(\alpha) = g(\beta).$$

E.g., $n = 9, r = 2, q = 13$;

$$\mathcal{A} = \{A_1 = \{1, 3, 9\}, A_2 = \{2, 6, 5\}, A_3 = \{4, 12, 10\}\},$$

Then $g(x) = x^3$ is constant on each of the A_i's

Construction of (n, k, r) LRC codes

Given $A \subset \mathbb{F}$, partition $\mathcal{A}$ into $(r+1)$-subsets.

To encode the message $a \in \mathbb{F}^k$, write $a = (a_{ij}, i = 0, \ldots, r-1; j = 0, ..., \frac{k}{r} - 1)$

Define the encoding polynomial

$$f_a(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{\frac{k}{r}-1} a_{ij} g(x)^j$$

A linear code $\mathcal{C}$ is constructed as follows:

$$\begin{aligned} Ev : & \mathbb{F}^k \to \mathbb{F}^n \\ & a \mapsto (f_a(\beta), \beta \in A) \end{aligned}$$

Construction of (n, k, r) LRC codes

Given $A \subset \mathbb{F}$, partition $\mathcal{A}$ into $(r+1)$-subsets.

To encode the message $a \in \mathbb{F}^k$, write $a = (a_{ij}, i = 0, \ldots, r-1; j = 0, ..., \frac{k}{r} - 1)$

Define the encoding polynomial

$$f_a(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{\frac{k}{r}-1} a_{ij} g(x)^j$$

A linear code $\mathcal{C}$ is constructed as follows:

$$\begin{aligned} Ev : & \mathbb{F}^k \to \mathbb{F}^n \\ & a \mapsto (f_a(\beta), \beta \in A) \end{aligned}$$

It is easy to show that the parameters of the constructed codes meet the Gopalan et al. bound with equality

I. Tamo and A.B., *A family of optimal locally recoverable codes*, T-IT August 2014

Constructing $g(x)$

Proposition

Let H be a subgroup of $\mathbb{F}_q^*$ or $\mathbb{F}_q^+$. The annihilator polynomial of H

$$g(x) = \prod_{h \in H} (x - h)$$

is constant on each coset of H.

Constructing $g(x)$

Proposition

Let H be a subgroup of $\mathbb{F}_q^*$ or $\mathbb{F}_q^+$. The annihilator polynomial of H

$$g(x) = \prod_{h \in H} (x - h)$$

is constant on each coset of H.

Further constructions:

J. Liu, S. Mesnager and L. Chen, *New constructions of optimal locally recoverable codes via good polynomials*, T-IT 2018

Summary of the construction

The optimal RS-like LRC codes are constructed as follows:

1. Take an RS code over $\mathbb{F}_q$ of length n and dimension $\frac{r+1}{r}k-2$
2. Isolate a carefully chosen k-dimensional subcode such the the polynomials become degree $r-1$ when restricted to recovering sets of size $r+1$.

Summary of the construction

The optimal RS-like LRC codes are constructed as follows:

1. Take an RS code over $\mathbb{F}_q$ of length n and dimension $\frac{r+1}{r}k-2$
2. Isolate a carefully chosen k-dimensional subcode such the the polynomials become degree $r-1$ when restricted to recovering sets of size $r+1$.

These codes are studied outside the storage context:

L. HOLZBAUR AND A. WACHTER-ZEH, *List decoding of locally repairable codes*, arXiv:1801.04229

A. MAZUMDAR, *Caoacity of locally repairable codes*, arXiv:1801.04229

S. KADHE AND R. CALDERBANK, *LRC codes with small availability*, arXiv:1701.02456

Generalization of the main construction

- The length of the constructed codes is limited to $n \leqslant q$

Generalization of the main construction

- The length of the constructed codes is limited to $n \leqslant q$
- To construct longer codes, we take a geometric point of view.

Generalization of the main construction

- The length of the constructed codes is limited to $n \leqslant q$
- To construct longer codes, we take a geometric point of view.
- Replace RS codes with codes on algebraic curves; it is possible to construct LRC codes of large n for a fixed q.

Generalization of the main construction

- The length of the constructed codes is limited to $n \leqslant q$
- To construct longer codes, we take a geometric point of view.
- Replace RS codes with codes on algebraic curves; it is possible to construct LRC codes of large n for a fixed q.
- Consider the set of pairs $(x, y) \in \mathbb{F}_9$ that satisfy the equation $x^3 + x = y^4$. There are 27 solutions, which give the evaluation set of points of size $n = 27$

Generalization of the main construction

- The length of the constructed codes is limited to $n \leqslant q$
- To construct longer codes, we take a geometric point of view.
- Replace RS codes with codes on algebraic curves; it is possible to construct LRC codes of large n for a fixed q.
- Consider the set of pairs $(x, y) \in \mathbb{F}_9$ that satisfy the equation $x^3 + x = y^4$. There are 27 solutions, which give the evaluation set of points of size $n = 27$
- We evaluate bivariate polynomials spanned by the monomials $V := \langle 1, y, y^2, x, xy, xy^2 \rangle$

Generalization of the main construction

- The length of the constructed codes is limited to $n \leqslant q$
- To construct longer codes, we take a geometric point of view.
- Replace RS codes with codes on algebraic curves; it is possible to construct LRC codes of large n for a fixed q.
- Consider the set of pairs $(x, y) \in \mathbb{F}_9$ that satisfy the equation $x^3 + x = y^4$. There are 27 solutions, which give the evaluation set of points of size $n = 27$
- We evaluate bivariate polynomials spanned by the monomials $V := \langle 1, y, y^2, x, xy, xy^2 \rangle$
- We obtain a 6-dimensional code with locality $r = 2$

$$\mathcal{C} : V \rightarrow \mathbb{F}_9^n$$

Generalization of the main construction

- The length of the constructed codes is limited to $n \leqslant q$
- To construct longer codes, we take a geometric point of view.
- Replace RS codes with codes on algebraic curves; it is possible to construct LRC codes of large n for a fixed q.
- Consider the set of pairs $(x, y) \in \mathbb{F}_9$ that satisfy the equation $x^3 + x = y^4$. There are 27 solutions, which give the evaluation set of points of size $n = 27$
- We evaluate bivariate polynomials spanned by the monomials $V := \langle 1, y, y^2, x, xy, xy^2 \rangle$
- We obtain a 6-dimensional code with locality $r = 2$

$$\mathcal{C} : V \to \mathbb{F}_9^n$$

- E.g., message $(1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5)$

$$F(x, y) = 1 + \alpha y + \alpha^2 y^2 + \alpha^3 x + \alpha^4 xy + \alpha^5 xy^2$$

$$F(0, 0) = 1 \text{ etc.}$$

A.B., I. Tamo, and S. Vlăduţ, *LRC codes on algebraic curves*, T-IT, Aug. 2017

Random LRC codes and a Gilbert-Varshamov type bound

Let $M(n, r, \delta n)$ be the max size of a code of length n, distance d, locality r

$$R(r, \delta) := \limsup_{n\to\infty} \frac{1}{n} \log M(n, r, \delta n)$$

GV-type bound:

$$R(r, \delta) \geqslant 1 - \min_{0<s\leqslant 1} \left\{ \frac{1}{r+1} \log_2((1+s)^{r+1} + (1-s)^{r+1}) - \delta \log_2 s \right\}.$$

Random LRC codes and a Gilbert-Varshamov type bound

Let $M(n, r, \delta n)$ be the max size of a code of length n, distance d, locality r

$$R(r,\delta) := \limsup_{n\to\infty} \frac{1}{n} \log M(n,r,\delta n)$$

GV-type bound:

$$R(r,\delta) \geqslant 1 - \min_{0<s\leqslant 1} \left\{ \frac{1}{r+1} \log_2((1+s)^{r+1} + (1-s)^{r+1}) - \delta \log_2 s \right\}.$$

Proof by random coding: Estimate the average weight enumerator for the ensemble given by

$$H = \left[\begin{array}{cccc} \boxed{11\ldots1} & & & \\ & \boxed{11\ldots1} & & \\ & & \ddots & \\ & & & \boxed{11\ldots1} \\ \hline & & H_L & \end{array} \right]$$

Random LRC codes and a Gilbert-Varshamov type bound

Let $M(n, r, \delta n)$ be the max size of a code of length n, distance d, locality r

$$R(r, \delta) := \limsup_{n\to\infty} \frac{1}{n} \log M(n, r, \delta n)$$

GV-type bound:

$$R(r, \delta) \geqslant 1 - \min_{0<s\leqslant 1} \left\{ \frac{1}{r+1} \log_2((1+s)^{r+1} + (1-s)^{r+1}) - \delta \log_2 s \right\}.$$

Proof by random coding: Estimate the average weight enumerator for the ensemble given by

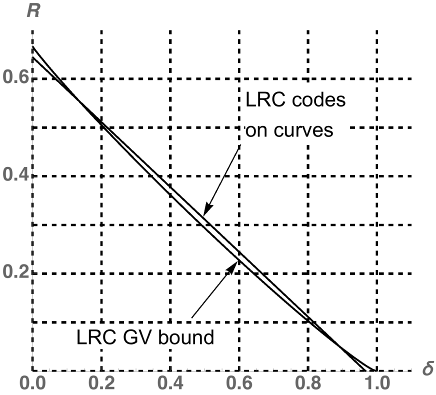
$$H = \left[\begin{array}{cccc} \boxed{11\ldots1} & & & \\ & \boxed{11\ldots1} & & \\ & & \ddots & \\ & & & \boxed{11\ldots1} \\ \hline & & H_L & \end{array}\right]$$

$$\Pr(\{d(\mathcal{C}) < \delta n\}) \leqslant \delta n q^{-n(\frac{r}{r+1} - R)} \min_{0<s\leqslant 1} \frac{b(s)^{\frac{n}{r+1}}}{s^{\delta n}}$$

H_L is a matrix with independent uniformly chosen elements of $\mathbb{F}_q$

V. Cadambe and A. Mazumdar, T-IT 2015; I. Tamo, A.B., and A. Frolov, T-IT, June 2016

Improving GV bound using LRC codes on curves



Locality $r = 2$, $q = 2^{10}$

A.B., I. Tamo, and S. Vlăduţ, *LRC codes on algebraic curves*, T-IT, Aug. 2017

More on bounds:

A. Agarwal et al., *Combinatorial alphabet-dependent bounds for locally recoverable codes*, T-IT 2018

Extensions

- Codes with availability
- Correcting $2, 3, \ldots$ erasures locally
- Hierarchical locality
- Maximally recoverable codes
- Maximum length of optimal LRC codes
- Cyclic LRC codes

Availability

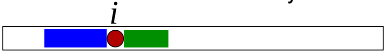
- "Hot data" accessed simultaneously by a very large number of users

Availability

- “Hot data” accessed simultaneously by a very large number of users
- Recovering an erasure from several disjoint repair groups increases the availability of the data.

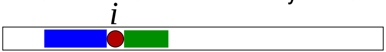
Availability

- "Hot data" accessed simultaneously by a very large number of users
- Recovering an erasure from several disjoint repair groups increases the availability of the data.
- Every coordinate is recoverable from the codeword symbols in several recovering sets:



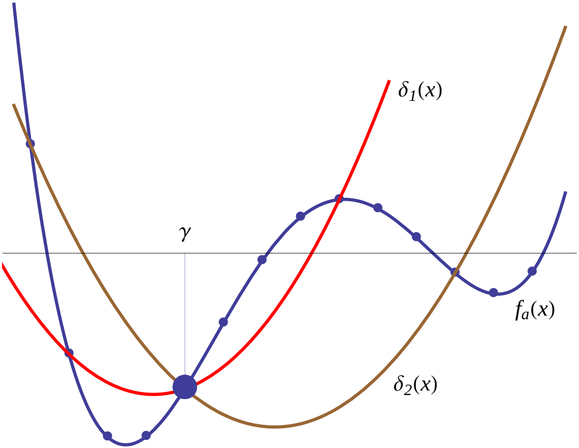
Availability

- "Hot data" accessed simultaneously by a very large number of users
- Recovering an erasure from several disjoint repair groups increases the availability of the data.
- Every coordinate is recoverable from the codeword symbols in several recovering sets:



- A code $\mathcal{C}$ is called an $LRC(2)$ code if every coordinate i has 2 disjoint recovering sets $R_{1,i}, |R_{1,i}| \leqslant r_1; R_{2,i}, |R_{2,i}| \leqslant r_2$

Multiple recovery sets: Idea of construction



$f_a(\gamma)$ can be found by interpolating $\delta_1(x)$ as well as $\delta_2(x)$

Multiple recovery sets: Example

Take $\mathbb{F} = \mathbb{F}_{13}$; $G, H \leqslant \mathbb{F}^*$; $G = \langle 5 \rangle, H = \langle 3 \rangle$

$$\mathcal{A}_G = \{\{1, 5, 12, 8\}, \{2, 10, 11, 3\}, \{4, 7, 9, 6\}\}$$

$$\mathcal{A}_H = \{\{1, 3, 9\}, \{2, 6, 5\}, \{4, 12, 10\}, \{7, 8, 11\}\}$$

Let

$$\mathbb{F}_{\mathcal{A}_G}[x] = \{f \in \mathbb{F}[x] : f \text{ is constant on } A_i, i = 1, 2, 3;\ \deg f < |\mathbb{F}^*|\}$$

$$\mathbb{F}_{\mathcal{A}_G}[x] = \langle 1, x^4, x^8 \rangle, \quad \mathbb{F}_{\mathcal{A}_H}[x] = \langle 1, x^3, x^6, x^9 \rangle$$

Multiple recovery sets: Example

Take $\mathbb{F} = \mathbb{F}_{13}$; $G, H \leqslant \mathbb{F}^*$; $G = \langle 5 \rangle, H = \langle 3 \rangle$

$$\mathcal{A}_G = \{\{1,5,12,8\},\{2,10,11,3\},\{4,7,9,6\}\}$$
$$\mathcal{A}_H = \{\{1,3,9\},\{2,6,5\},\{4,12,10\},\{7,8,11\}\}$$

Let

$$\mathbb{F}_{\mathcal{A}_G}[x] = \{f \in \mathbb{F}[x] : f \text{ is constant on } A_i, i = 1,2,3;\ \deg f < |\mathbb{F}^*|\}$$
$$\mathbb{F}_{\mathcal{A}_G}[x] = \langle 1, x^4, x^8 \rangle, \quad \mathbb{F}_{\mathcal{A}_H}[x] = \langle 1, x^3, x^6, x^9 \rangle$$

We construct an LRC $(12, 4, \{2,3\})$, distance $\geqslant 6$, code $\mathcal{C} : \mathbb{F}^4 \to \mathbb{F}^{12}$

$$a = (a_0, a_1, a_2, a_3) \mapsto f_a(x) = a_0 + a_1 x + a_2 x^4 + a_3 x^6$$

$$f_a(x) = \sum_{i=0}^{2} f_i(x) x^i, \text{ where } f_0(x) = a_0 + a_2 x^4,\ f_1(x) = a_1,\ f_2(x) = a_3 x^4; f_i \in \mathbb{F}_{\mathcal{A}}[x]$$

$$f_a(x) = \sum_{j=0}^{1} g_j(x) x^j \text{ where } g_0(x) = a_0 + a_3 x^6, g_1(x) = a_1 + a_2 x^3; g_j \in \mathbb{F}_{\mathcal{A}_H}[x]$$

E.g., $f_a(1)$ can be recovered by computing $\delta_1(x), x \in \{5, 12, 8\}$ OR $\delta_2(x), x \in \{3, 9\}$

Other constructions

- Product codes
- Codes on bipartite graphs
- Direct-sum codes

Open problem: Bounds on codes with availability

- Known bounds:

  Let $\mathcal{C}$ be an (n,k,r,t) LRC code with t disjoint recovering sets of size r. Then the rate of $\mathcal{C}$ satisfies

$$\frac{k}{n} \leqslant \frac{1}{\prod_{j=1}^{t}(1+\frac{1}{jr})}$$

  The minimum distance of $\mathcal{C}$ is bounded above as follows:

$$d \leqslant n - \sum_{i=0}^{t} \left\lfloor \frac{k-1}{r^i} \right\rfloor.$$

I. Tamo, A.B., and A Frolov, *Bounds on the Parameters of Locally Recoverable Codes,* T-IT 2016

$$d \leqslant n-k+2-\left\lceil \frac{t(k-1)+1}{t(r-1)+1} \right\rceil$$

A. Wang and Z. Zhang, *Repair locality with multiple erasure tolerance,* T-IT 2014

More on bounds:

N. Prakash, V. Lalitha, and P. V. Kumar, *Codes with locality for two erasures*, ISIT 2014
S. B. Balaji and P. V. Kumar, *Bounds on ... codes with availability*, ISIT 2017 (improved results for linear codes)

- The RS-like construction can be extended to $t \geqslant 2$ recovering sets, but the resulting codes are not known to be optimal

Remarks on the bounds, and Graph-theoretic connections

- The bound on the rate of codes with availability t can be simplified:

$$\frac{k}{n} \leqslant \frac{1}{\sqrt[r]{t+1}}$$

- Tighter bounds are available in some cases (BALAJI-KUMAR, arXiv1611.00159)

$$R(r,2) \leqslant \frac{r}{r+2}, \; R(r,3) \leqslant \frac{r^2}{(r+1)^2}$$

- Problems related to multiplicities, e.g., availability or sequential repair, often can be interpreted in terms of graph theory or matroid theory.
- To derive the bounds, we note that multiple repair groups create dependence relations on the set of coordinates; we analyze the "expansion" of dependencies in the recovering graph as we add vertices successively.

M. GRETZEL AND C. HOLLANTI, *The complete hierarchical locality of the punctured simplex code*, arXiv:1901.03149

R. FREIJ-HOLLANTI, C. HOLLANTI, AND T. WESTERBCK, Matroid theory and storage codes, arXiv:1704.04007

Correcting $\geqslant 2$ erasures locally

In the event that more than one node in the encoding have failed, we need to correct more than one erasure locally

A code $\mathcal{C}$ is said to have the (ρ, r) locality property if each coordinate i is contained in a subset $A_i \subset [n], |A_i| \leqslant r + \rho - 1$ such that the restriction C_{A_i} forms a code of distance $\geqslant \rho$.

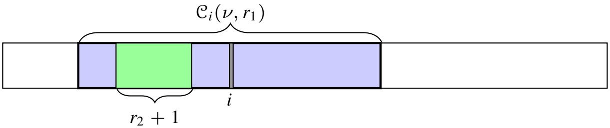
The distance of the code $\mathcal{C}$ satisfies the bound

$$d \leqslant n - k + 1 - \left(\left\lceil \frac{k}{r} \right\rceil - 1 \right) (\rho - 1)$$

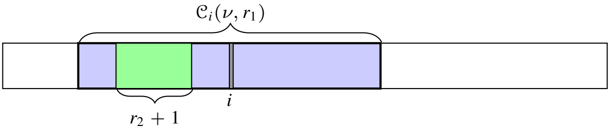
G.M. Kamath et al., *Codes with local regeneration and erasure correction*, T-IT. Aug. 2014

The RS-like construction can be extended to this case, the parameters of the resulting codes meet this bound

Hierarchical locality



Hierarchical locality

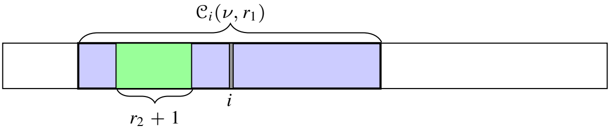


Every coordinate i is in a code $\mathcal{C}_i$ that

- corrects several erasures (distance $\geqslant \rho_1$)
- is LRC

B. Sasidharan et al., *Codes with hierarchical locality*, ISIT 2015

Hierarchical locality



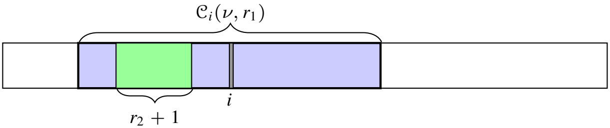
Every coordinate i is in a code $\mathcal{C}_i$ that

- corrects several erasures (distance $\geqslant \rho_1$)
- is LRC

B. Sasidharan et al., *Codes with hierarchical locality*, ISIT 2015

The Gopalan et al. bound can be extended to local codes with hierarchy (Sasidharan e.a.)

Hierarchical locality



Every coordinate i is in a code $\mathcal{C}_i$ that

- corrects several erasures (distance $\geqslant \rho_1$)
- is LRC

B. Sasidharan et al., *Codes with hierarchical locality*, ISIT 2015

The Gopalan et al. bound can be extended to local codes with hierarchy (Sasidharan e.a.)

Constructions of optimal RS-type codes and of codes on algebraic curves

S. Ballentine et al., *Codes with hierarchical locality from covering maps of curves*, T-IT 2019

Open problem: Maximally recoverable codes

- A code is MDS if any k symbols suffice to recover the data

Open problem: Maximally recoverable codes

Locality and efficient data retrieval

- A code is MDS if any k symbols suffice to recover the data
- LRC codes are not MDS because k symbols that contain a recovering set cannot be used to recover the data

Open problem: Maximally recoverable codes

- A code is MDS if any k symbols suffice to recover the data
- LRC codes are not MDS because k symbols that contain a recovering set cannot be used to recover the data
- Problem: Construct LRC codes that are as close to MDS as possible

Open problem: Maximally recoverable codes

Locality and efficient data retrieval

- A code is MDS if any k symbols suffice to recover the data
- LRC codes are not MDS because k symbols that contain a recovering set cannot be used to recover the data
- Problem: Construct LRC codes that are as close to MDS as possible

> A code is called maximally recoverable if any k-tuple of coordinates that does not contain a local constraint, has full rank.
>
> M. Chen, C. Huang, and J. Li, ISIT 2007; P. Gopalan et al., T-IT 2010; 2014

-

Open problem: Maximally recoverable codes

- A code is MDS if any k symbols suffice to recover the data
- LRC codes are not MDS because k symbols that contain a recovering set cannot be used to recover the data
- Problem: Construct LRC codes that are as close to MDS as possible

A code is called maximally recoverable if any k-tuple of coordinates that does not contain a local constraint, has full rank.

- M. Chen, C. Huang, and J. Li, ISIT 2007; P. Gopalan et al., T-IT 2010; 2014

- Rephrased, if $B \subset [n]$ is a subset such that $|B| \geqslant k$ and B does not contain a local constraint, the restriction $\mathcal{C}|_B$ is an MDS code

- A code is MDS if any k symbols suffice to recover the data
- LRC codes are not MDS because k symbols that contain a recovering set cannot be used to recover the data
- Problem: Construct LRC codes that are as close to MDS as possible

> A code is called maximally recoverable if any k-tuple of coordinates that does not contain a local constraint, has full rank.
>
> M. Chen, C. Huang, and J. Li, ISIT 2007; P. Gopalan et al., T-IT 2010; 2014

- Rephrased, if $B \subset [n]$ is a subset such that $|B| \geqslant k$ and B does not contain a local constraint, the restriction $\mathcal{C}|_B$ is an MDS code
- It is not difficult to prove that MR codes exist, but the underlying finite field is of large size $q \geqslant \binom{n}{k}$.

Open problem: Maximally recoverable codes

- A code is MDS if any k symbols suffice to recover the data
- LRC codes are not MDS because k symbols that contain a recovering set cannot be used to recover the data
- Problem: Construct LRC codes that are as close to MDS as possible
-
  A code is called maximally recoverable if any k-tuple of coordinates that does not contain a local constraint, has full rank.

  M. Chen, C. Huang, and J. Li, ISIT 2007; P. Gopalan et al., T-IT 2010; 2014
- Rephrased, if $B \subset [n]$ is a subset such that $|B| \geqslant k$ and B does not contain a local constraint, the restriction $\mathcal{C}|_B$ is an MDS code
- It is not difficult to prove that MR codes exist, but the underlying finite field is of large size $q \geqslant \binom{n}{k}$.
- Construction of MR codes over small fields and bounds on the field size form a difficult open problem

Open problem: Maximally recoverable codes

- A code is MDS if any k symbols suffice to recover the data
- LRC codes are not MDS because k symbols that contain a recovering set cannot be used to recover the data
- Problem: Construct LRC codes that are as close to MDS as possible

> A code is called maximally recoverable if any k-tuple of coordinates that does not contain a local constraint, has full rank.
>
> M. Chen, C. Huang, and J. Li, ISIT 2007; P. Gopalan et al., T-IT 2010; 2014

- Rephrased, if $B \subset [n]$ is a subset such that $|B| \geqslant k$ and B does not contain a local constraint, the restriction $\mathcal{C}|_B$ is an MDS code
- It is not difficult to prove that MR codes exist, but the underlying finite field is of large size $q \geqslant \binom{n}{k}$.
- Construction of MR codes over small fields and bounds on the field size form a difficult open problem
- Partial MDS codes - array configuration

M. Blaum et al.,*Partial MDS codes and their application to RAID type of architectures*, T-IT 2013

MR codes are optimal

Lemma

MR codes are optimal LRC codes

MR codes are optimal

Lemma

MR codes are optimal LRC codes

Proof:

MR codes are optimal

Lemma

MR codes are optimal LRC codes

Proof:

- Assume r divides k

MR codes are optimal

Lemma

MR codes are optimal LRC codes

Proof:

- Assume r divides k
- MR code is an Optimal LRC code

MR codes are optimal

Lemma

MR codes are optimal LRC codes

Proof:

- Assume r divides k
- MR code is an Optimal LRC code

$\Leftrightarrow d = n - k - \frac{k}{r} + 2$

MR codes are optimal

Lemma

MR codes are optimal LRC codes

Proof:

- Assume r divides k
- MR code is an Optimal LRC code

$\Leftrightarrow d = n - k - \frac{k}{r} + 2$

$\Leftrightarrow$ any $d - 1 = n - k - \frac{k}{r} + 1$ erasures are recoverable

MR codes are optimal

Lemma

MR codes are optimal LRC codes

Proof:

- Assume r divides k
- MR code is an Optimal LRC code

$\Leftrightarrow d = n - k - \frac{k}{r} + 2$

$\Leftrightarrow$ any $d - 1 = n - k - \frac{k}{r} + 1$ erasures are recoverable

$\Leftrightarrow$ any $n - (d - 1) = k + \frac{k}{r} - 1$ coordinates suffice for decoding

MR codes are optimal

Lemma

MR codes are optimal LRC codes

Proof:

- Assume r divides k
- MR code is an Optimal LRC code

  $\Leftrightarrow d = n - k - \frac{k}{r} + 2$

  $\Leftrightarrow$ any $d - 1 = n - k - \frac{k}{r} + 1$ erasures are recoverable

  $\Leftrightarrow$ any $n - (d - 1) = k + \frac{k}{r} - 1$ coordinates suffice for decoding
- Any $k + \frac{k}{r} - 1$ coordinates contain a subset S s.t.

MR codes are optimal

Lemma

MR codes are optimal LRC codes

Proof:

- Assume r divides k
- MR code is an Optimal LRC code

  $\Leftrightarrow d = n - k - \frac{k}{r} + 2$

  $\Leftrightarrow$ any $d - 1 = n - k - \frac{k}{r} + 1$ erasures are recoverable

  $\Leftrightarrow$ any $n - (d-1) = k + \frac{k}{r} - 1$ coordinates suffice for decoding
- Any $k + \frac{k}{r} - 1$ coordinates contain a subset S s.t.
  1. $|S| = k$

MR codes are optimal

Lemma

MR codes are optimal LRC codes

Proof:

- Assume r divides k
- MR code is an Optimal LRC code

  $\Leftrightarrow d = n - k - \frac{k}{r} + 2$

  $\Leftrightarrow$ any $d - 1 = n - k - \frac{k}{r} + 1$ erasures are recoverable

  $\Leftrightarrow$ any $n - (d - 1) = k + \frac{k}{r} - 1$ coordinates suffice for decoding

- Any $k + \frac{k}{r} - 1$ coordinates contain a subset S s.t.
  1. $|S| = k$
  2. $\forall i, \mathcal{R}_i \nsubseteq S$

MR codes are optimal

Lemma

MR codes are optimal LRC codes

Proof:

- Assume r divides k
- MR code is an Optimal LRC code

$\Leftrightarrow d = n - k - \frac{k}{r} + 2$

$\Leftrightarrow$ any $d - 1 = n - k - \frac{k}{r} + 1$ erasures are recoverable

$\Leftrightarrow$ any $n - (d - 1) = k + \frac{k}{r} - 1$ coordinates suffice for decoding

- Any $k + \frac{k}{r} - 1$ coordinates contain a subset S s.t.
  1. $|S| = k$
  2. $\forall i, \mathcal{R}_i \nsubseteq S$
- By the MR property, S suffices for decoding

MR codes are optimal

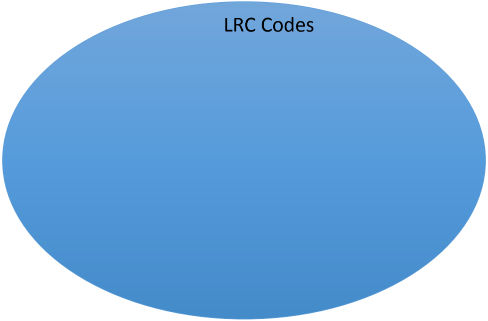
Lemma

MR codes are optimal LRC codes

MR codes are optimal

Lemma

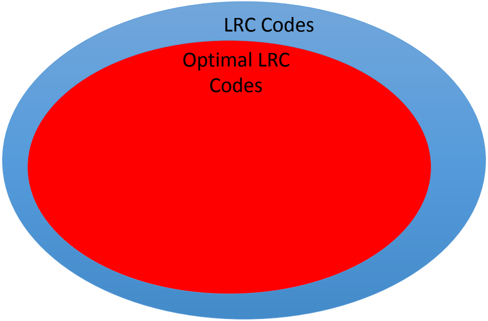
MR codes are optimal LRC codes



MR codes are optimal

Lemma

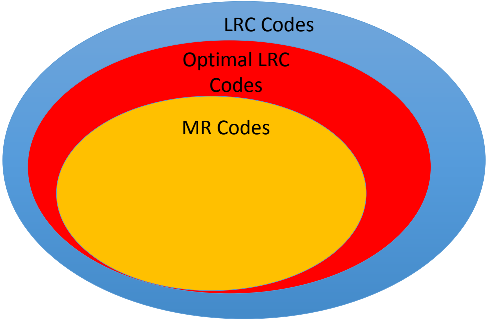
MR codes are optimal LRC codes



MR codes are optimal

Lemma

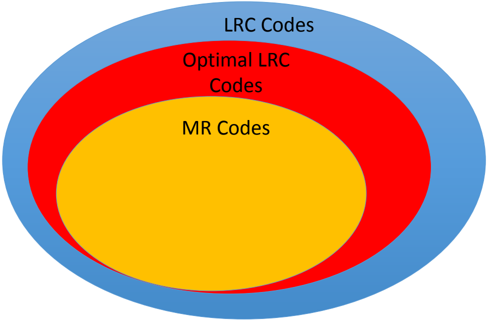
MR codes are optimal LRC codes



MR codes are optimal

Lemma

MR codes are optimal LRC codes

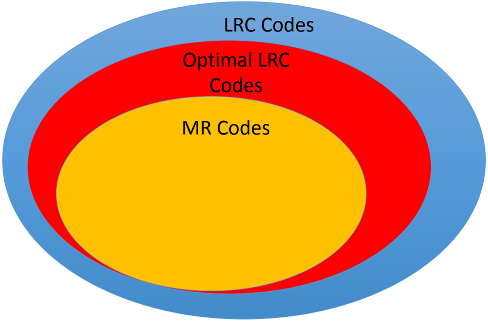


Q: MR codes = Optimal LRC codes ?

MR codes are optimal

Lemma

MR codes are optimal LRC codes



Q: MR codes = Optimal LRC codes ? Ans: No

MR codes through linearized polynomials

A. S. RAWAT ET AL.,,*Optimal locally repairable and secure codes for distributed storage systems*, T-IT 2014

MR codes through linearized polynomials

A. S. RAWAT ET AL.,,*Optimal locally repairable and secure codes for distributed storage systems*, T-IT 2014

- Flexible set of parameters n, k, r

MR codes through linearized polynomials

A. S. RAWAT ET AL.,,*Optimal locally repairable and secure codes for distributed storage systems*, T-IT 2014

- Flexible set of parameters n, k, r ✓

MR codes through linearized polynomials

A. S. RAWAT ET AL.,,*Optimal locally repairable and secure codes for distributed storage systems*, T-IT 2014

- Flexible set of parameters n, k, r ✓

- Need $m = \frac{nr}{r+1}$ linearly independent elements over $\mathbb{F}_2$

MR codes through linearized polynomials

A. S. RAWAT ET AL.,,*Optimal locally repairable and secure codes for distributed storage systems*, T-IT 2014

- Flexible set of parameters n, k, r ✓
- Need $m = \frac{nr}{r+1}$ linearly independent elements over $\mathbb{F}_2 \implies |\mathbb{F}| = 2^{\frac{nr}{r+1}}$

MR codes through linearized polynomials

A. S. Rawat et al.,,*Optimal locally repairable and secure codes for distributed storage systems*, T-IT 2014

- Flexible set of parameters n, k, r ✓
- Need $m = \frac{nr}{r+1}$ linearly independent elements over $\mathbb{F}_2 \implies |\mathbb{F}| = 2^{\frac{nr}{r+1}}$
- Field size is exponential in n

MR codes through linearized polynomials

A. S. RAWAT ET AL.,,*Optimal locally repairable and secure codes for distributed storage systems*, T-IT 2014

- Flexible set of parameters n, k, r ✓
- Need $m = \frac{nr}{r+1}$ linearly independent elements over $\mathbb{F}_2 \implies |\mathbb{F}| = 2^{\frac{nr}{r+1}}$
- Field size is exponential in n ×

MR codes through linearized polynomials

A. S. RAWAT ET AL.,,*Optimal locally repairable and secure codes for distributed storage systems*, T-IT 2014

- Flexible set of parameters n, k, r ✓
- Need $m = \frac{nr}{r+1}$ linearly independent elements over $\mathbb{F}_2 \implies |\mathbb{F}| = 2^{\frac{nr}{r+1}}$
- Field size is exponential in n ×
- Can we do better?

U. MARTINEZ-PEÑAS AND F. KSCHISCHANG, *Universal and dynamic locally repairable codes with maximal recoverability via sum-rank codes*, arXiv:1809.11158

Open problem: Maximal length of Opt-LRC codes

- A code is called Opt-LRC if its distance d is maximum possible:

$$d = n - k - \left\lceil \frac{k}{r} \right\rceil + 2$$

Open problem: Maximal length of Opt-LRC codes

- A code is called Opt-LRC if its distance d is maximum possible:

$$d = n - k - \left\lceil \frac{k}{r} \right\rceil + 2$$

- Without locality:

Every code has locality k (EXERCISE: why?), and then $d \leqslant n-k+1$. If $=$, then the code is called MDS. The maximum length of a q-ary MDS code is conjectured to be $q+2$.

The MDS conjecture is a famous open problem (latest advances by SIMEON BALL, 2012)

Open problem: Maximal length of Opt-LRC codes

- A code is called Opt-LRC if its distance d is maximum possible:

$$d = n - k - \left\lceil \frac{k}{r} \right\rceil + 2$$

- Without locality:

Every code has locality k (EXERCISE: why?), and then $d \leqslant n - k + 1$. If $=$, then the code is called MDS. The maximum length of a q-ary MDS code is conjectured to be $q + 2$.

The MDS conjecture is a famous open problem (latest advances by SIMEON BALL, 2012)

- With locality: The length of an Opt-LRC code with $d \geqslant 5$ is

$$n \leqslant \frac{d-1}{2q-2} q^{4+\frac{1}{d}} + \frac{r+1}{r}.$$

V. GURUSWAMI ET AL., *How long can optimal locally recoverable codes be?*, T-IT 2019

Open problem: Maximal length of Opt-LRC codes

- A code is called Opt-LRC if its distance d is maximum possible:

$$d = n - k - \left\lceil \frac{k}{r} \right\rceil + 2$$

- Without locality:

Every code has locality k (EXERCISE: why?), and then $d \leqslant n-k+1$. If $=$, then the code is called MDS. The maximum length of a q-ary MDS code is conjectured to be $q+2$.

The MDS conjecture is a famous open problem (latest advances by SIMEON BALL, 2012)

- With locality: The length of an Opt-LRC code with $d \geqslant 5$ is

$$n \leqslant \frac{d-1}{2q-2} q^{4+\frac{1}{d}} + \frac{r+1}{r}.$$

V. GURUSWAMI ET AL., *How long can optimal locally recoverable codes be?*, T-IT 2019

- LUO-XING-YUAN, T-IT 2019: Opt-LRC codes of distance $3, 4$ and unbounded length

Open problem: Maximal length of Opt-LRC codes

- A code is called Opt-LRC if its distance d is maximum possible:

$$d = n - k - \left\lceil \frac{k}{r} \right\rceil + 2$$

- Without locality:

Every code has locality k (EXERCISE: why?), and then $d \leqslant n - k + 1$. If $=$, then the code is called MDS. The maximum length of a q-ary MDS code is conjectured to be $q + 2$.

The MDS conjecture is a famous open problem (latest advances by SIMEON BALL, 2012)

- With locality: The length of an Opt-LRC code with $d \geqslant 5$ is

$$n \leqslant \frac{d-1}{2q-2} q^{4+\frac{1}{d}} + \frac{r+1}{r}.$$

V. GURUSWAMI ET AL., *How long can optimal locally recoverable codes be?*, T-IT 2019

- LUO-XING-YUAN, T-IT 2019: Opt-LRC codes of distance $3, 4$ and unbounded length
- L.JIN, T-IT 2019: Opt-LRC codes of length q^2 and distance $5, 6$

Further variants of the repair problem

- Sequential repair: For a subset $c_{i_1}, \ldots, c_{i_t}$ of t erased nodes, it is possible to find a repair group of size $\leqslant r$ to recover c_{i_1}, then a repair group of size $\leqslant r$ (possibly including c_{i_1}) that recovers c_{i_2}, then another repair group of size $\leqslant r$ for c_{i_3}, etc.

  N. Prakash, V. Lalitha, and P. V. Kumar, *Codes with locality for two erasures*, ISIT2014

  S. B. Balaji, G. R. Kini and P. V. Kumar, *A tight lower bound...*, arXiv:1812.02501

Further variants of the repair problem

- Sequential repair: For a subset $c_{i_1}, \dots, c_{i_t}$ of t erased nodes, it is possible to find a repair group of size $\leqslant r$ to recover c_{i_1}, then a repair group of size $\leqslant r$ (possibly including c_{i_1}) that recovers c_{i_2}, then another repair group of size $\leqslant r$ for c_{i_3}, etc.

  N. PRAKASH, V. LALITHA, AND P. V. KUMAR, *Codes with locality for two erasures*, ISIT2014

  S. B. BALAJI, G. R. KINI AND P. V. KUMAR, *A tight lower bound...*, arXiv:1812.02501

- Parallel repair: Same as sequential, but the repaired symbols are not used to recover subsequent erasures

Further variants of the repair problem

- **Sequential repair:** For a subset $c_{i_1}, \dots, c_{i_t}$ of t erased nodes, it is possible to find a repair group of size $\leqslant r$ to recover c_{i_1}, then a repair group of size $\leqslant r$ (possibly including c_{i_1}) that recovers c_{i_2}, then another repair group of size $\leqslant r$ for c_{i_3}, etc.

  N. PRAKASH, V. LALITHA, AND P. V. KUMAR, *Codes with locality for two erasures*, ISIT2014

  S. B. BALAJI, G. R. KINI AND P. V. KUMAR, *A tight lower bound...*, arXiv:1812.02501

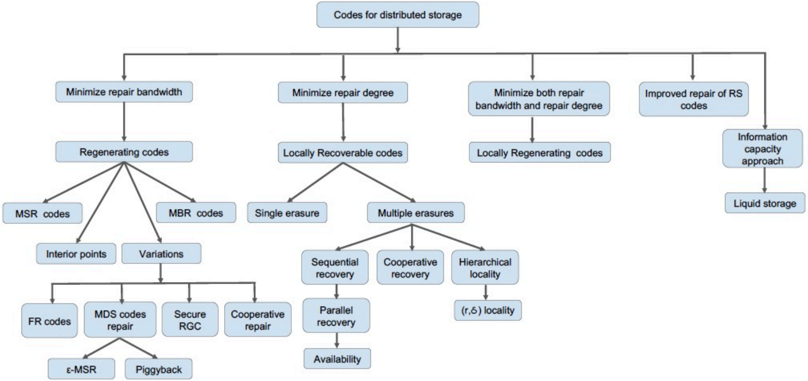
- **Parallel repair:** Same as sequential, but the repaired symbols are not used to recover subsequent erasures

- **Cooperative repair**

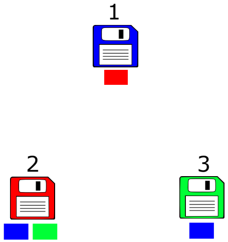
  S. KADHE ET AL., *On an Equivalence Between Single-Server PIR with Side Information and Locally Recoverable Codes*, arXiv:1907.00598

S.B. BALAJI et al., *Erasure coding for distributed storage: An overview*, arXiv:1806.04437

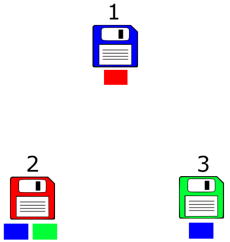
Different versions of the repair problem



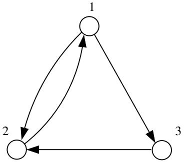
LRC codes on graphs [Mazumdar 2014, Shanmugam and Dimakis 2014]



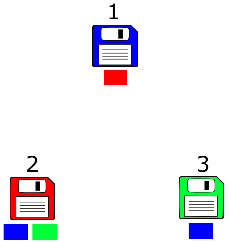
LRC codes on graphs [Mazumdar 2014, Shanmugam and Dimakis 2014]



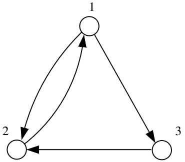
- Storage recovery graph G



LRC codes on graphs [Mazumdar 2014, Shanmugam and Dimakis 2014]

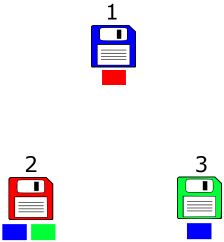


- Storage recovery graph G

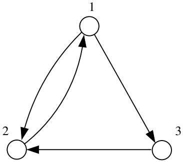


- Each node can recover its content from its incoming neighbors

LRC codes on graphs [Mazumdar 2014, Shanmugam and Dimakis 2014]

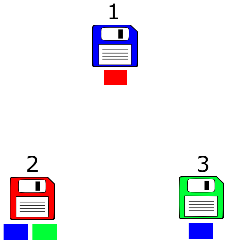


- Storage recovery graph G

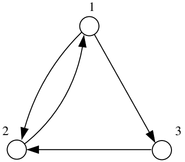


- Each node can recover its content from its incoming neighbors
- Recovery sets:

LRC codes on graphs [Mazumdar 2014, Shanmugam and Dimakis 2014]

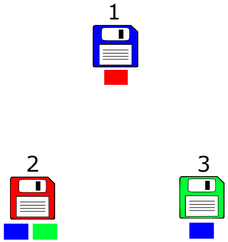


- Storage recovery graph G

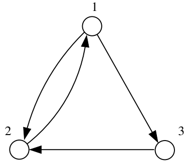


- Each node can recover its content from its incoming neighbors
- Recovery sets: $A_1 = \{2\}$,

LRC codes on graphs [Mazumdar 2014, Shanmugam and Dimakis 2014]

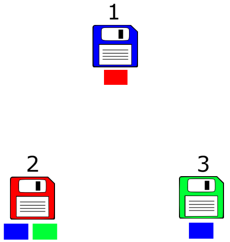


- Storage recovery graph G

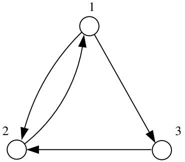


- Each node can recover its content from its incoming neighbors
- Recovery sets: $A_1 = \{2\}$, $A_2 = \{1, 3\}$, $A_3 = \{1\}$

LRC codes on graphs [Mazumdar 2014, Shanmugam and Dimakis 2014]

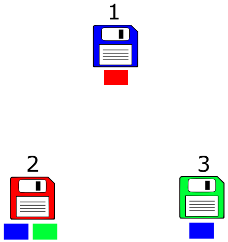


- Storage recovery graph G

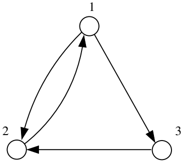


- Each node can recover its content from its incoming neighbors
- Recovery sets: $A_1 = \{2\}$, $A_2 = \{1, 3\}$, $A_3 = \{1\}$
- How much data can be stored?

LRC codes on graphs [Mazumdar 2014, Shanmugam and Dimakis 2014]



- Storage recovery graph G



- Each node can recover its content from its incoming neighbors
- Recovery sets: $A_1 = \{2\}$, $A_2 = \{1, 3\}$, $A_3 = \{1\}$
- How much data can be stored?
- Which coding scheme achieves the limit?

Storage Capacity

- The network is modeled by a (directed) graph $G = (V, E)$, $|V| = n$

Storage Capacity

- The network is modeled by a (directed) graph $G = (V, E)$, $|V| = n$
- Each node (vertex) stores a symbol from $\mathbb{F}_q$

Storage Capacity

- The network is modeled by a (directed) graph $G = (V, E)$, $|V| = n$
- Each node (vertex) stores a symbol from $\mathbb{F}_q$
- Storage code:

Storage Capacity

- The network is modeled by a (directed) graph $G = (V, E)$, $|V| = n$
- Each node (vertex) stores a symbol from $\mathbb{F}_q$
- Storage code:
  1. A set of vectors $\mathcal{C} \subseteq \mathbb{F}_q^n$

Storage Capacity

- The network is modeled by a (directed) graph $G = (V, E)$, $|V| = n$
- Each node (vertex) stores a symbol from $\mathbb{F}_q$
- Storage code:
  1. A set of vectors $\mathcal{C} \subseteq \mathbb{F}_q^n$
  2. n recovery functions f_i, s.t. for any $(x_1, \ldots, x_n) \in \mathcal{C}$

Storage Capacity

- The network is modeled by a (directed) graph $G = (V, E)$, $|V| = n$
- Each node (vertex) stores a symbol from $\mathbb{F}_q$
- Storage code:
  1. A set of vectors $\mathcal{C} \subseteq \mathbb{F}_q^n$
  2. n recovery functions f_i, s.t. for any $(x_1, \ldots, x_n) \in \mathcal{C}$

$$f_i(x_j : j \in N(i)) = x_i$$

Storage Capacity

- The network is modeled by a (directed) graph $G = (V, E)$, $|V| = n$
- Each node (vertex) stores a symbol from $\mathbb{F}_q$
- Storage code:
  1. A set of vectors $\mathcal{C} \subseteq \mathbb{F}_q^n$
  2. n recovery functions f_i, s.t. for any $(x_1, \dots, x_n) \in \mathcal{C}$
  $$f_i(x_j : j \in N(i)) = x_i$$
- The storage capacity of G over $\mathbb{F}_q$

$$\mathrm{Cap}_q(G) = \max_{\substack{\mathcal{C}\subseteq\mathbb{F}_q^n \text{ is a} \\ \text{storage code for } G}} \log_q |\mathcal{C}|$$

Storage Capacity

- The network is modeled by a (directed) graph $G = (V, E)$, $|V| = n$
- Each node (vertex) stores a symbol from $\mathbb{F}_q$
- Storage code:
  1. A set of vectors $\mathcal{C} \subseteq \mathbb{F}_q^n$
  2. n recovery functions f_i, s.t. for any $(x_1, \ldots, x_n) \in \mathcal{C}$

$$f_i(x_j : j \in N(i)) = x_i$$

- The storage capacity of G over $\mathbb{F}_q$

$$\mathrm{Cap}_q(G) = \max_{\substack{\mathcal{C} \subseteq \mathbb{F}_q^n \text{ is a} \\ \text{storage code for } G}} \log_q |\mathcal{C}| \leqslant n$$

Storage Capacity

- The network is modeled by a (directed) graph $G = (V, E)$, $|V| = n$
- Each node (vertex) stores a symbol from $\mathbb{F}_q$
- Storage code:
  1. A set of vectors $\mathcal{C} \subseteq \mathbb{F}_q^n$
  2. n recovery functions f_i, s.t. for any $(x_1, \ldots, x_n) \in \mathcal{C}$

$$f_i(x_j : j \in N(i)) = x_i$$

- The storage capacity of G over $\mathbb{F}_q$

$$\mathsf{Cap}_q(G) = \max_{\substack{\mathcal{C} \subseteq \mathbb{F}_q^n \text{ is a} \\ \text{storage code for } G}} \log_q |\mathcal{C}| \leqslant n$$

- The storage capacity of G is

$$\mathsf{Cap}(G) = \sup_q \mathsf{Cap}_q(G)$$

Storage Capacity

- The network is modeled by a (directed) graph $G = (V, E)$, $|V| = n$
- Each node (vertex) stores a symbol from $\mathbb{F}_q$
- Storage code:
  1. A set of vectors $\mathcal{C} \subseteq \mathbb{F}_q^n$
  2. n recovery functions f_i, s.t. for any $(x_1, \ldots, x_n) \in \mathcal{C}$

$$f_i(x_j : j \in N(i)) = x_i$$

- The storage capacity of G over $\mathbb{F}_q$

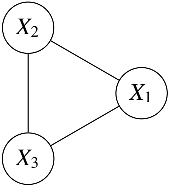
$$\mathsf{Cap}_q(G) = \max_{\substack{\mathcal{C} \subseteq \mathbb{F}_q^n \text{ is a} \\ \text{storage code for } G}} \log_q |\mathcal{C}| \leqslant n$$

- The storage capacity of G is

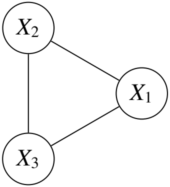
$$\mathsf{Cap}(G) = \sup_q \mathsf{Cap}_q(G) = \lim_{q \to \infty} \mathsf{Cap}_q(G) \text{ (Fekete's lemma)}$$

LRC codes on graphs - Example

LRC codes on graphs - Example

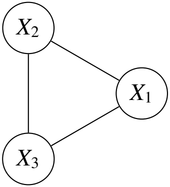


LRC codes on graphs - Example



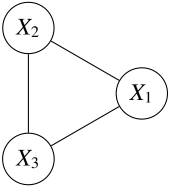
- Each node stores a single bit

LRC codes on graphs - Example



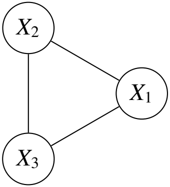
- Each node stores a single bit
- Two bits of information b_1, b_2 can be stored $(\text{Cap}_2(G) = 2)$

LRC codes on graphs - Example



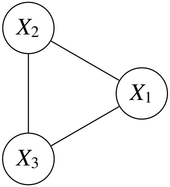
- Each node stores a single bit
- Two bits of information b_1, b_2 can be stored $(\mathrm{Cap}_2(G) = 2)$
- Store:
  - $X_1 = b_1$
  - $X_2 = b_2$
  - $X_3 = b_1 + b_2$

LRC codes on graphs - Example



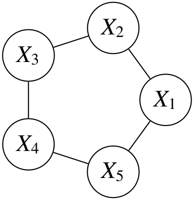
- Each node stores a single bit
- Two bits of information b_1, b_2 can be stored $(\mathrm{Cap}_2(G) = 2)$
- Store:
  - $X_1 = b_1$
  - $X_2 = b_2$
  - $X_3 = b_1 + b_2$
- $\mathrm{Cap}(G) = \sup_q \mathrm{Cap}_q(G)$

LRC codes on graphs - Example

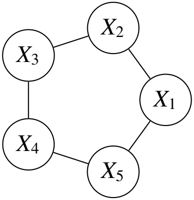


- Each node stores a single bit
- Two bits of information b_1, b_2 can be stored $(\mathrm{Cap}_2(G) = 2)$
- Store:
  - $X_1 = b_1$
  - $X_2 = b_2$
  - $X_3 = b_1 + b_2$
- $\mathrm{Cap}(G) = \sup_q \mathrm{Cap}_q(G) = 2$

LRC codes on graphs - Example

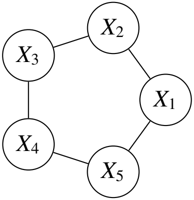


LRC codes on graphs - Example



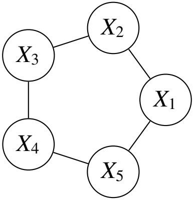
- $\mathcal{C} = \{(0,0,0,0,0),(0,1,1,0,0),(0,0,0,1,1),(1,1,0,1,1),(1,1,1,0,1)\}$

LRC codes on graphs - Example



- $\mathcal{C} = \{(0,0,0,0,0),(0,1,1,0,0),(0,0,0,1,1),(1,1,0,1,1),(1,1,1,0,1)\}$
- $X_1 = X_2 \wedge X_5,$

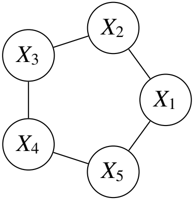
LRC codes on graphs - Example



- $\mathcal{C} = \{(0,0,0,0,0), (0,1,1,0,0), (0,0,0,1,1), (1,1,0,1,1), (1,1,1,0,1)\}$

- $X_1 = X_2 \wedge X_5, X_2 = X_1 \vee X_3, X_3 = X_2 \wedge \overline{X_4}, X_4 = \overline{X_3} \wedge X_5, X_5 = X_1 \vee X_4$

LRC codes on graphs - Example

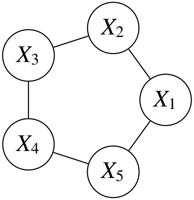


- $\mathcal{C} = \{(0,0,0,0,0),(0,1,1,0,0),(0,0,0,1,1),(1,1,0,1,1),(1,1,1,0,1)\}$

- $X_1 = X_2 \wedge X_5, X_2 = X_1 \vee X_3, X_3 = X_2 \wedge \overline{X_4}, X_4 = \overline{X_3} \wedge X_5, X_5 = X_1 \vee X_4$

- $\mathsf{Cap}_2(G) \geqslant \log_2(5) = 2.32...$

LRC codes on graphs - Example



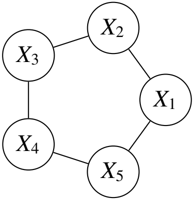
- $\mathcal{C} = \{(0,0,0,0,0), (0,1,1,0,0), (0,0,0,1,1), (1,1,0,1,1), (1,1,1,0,1)\}$

- $X_1 = X_2 \wedge X_5, X_2 = X_1 \vee X_3, X_3 = X_2 \wedge \overline{X_4}, X_4 = \overline{X_3} \wedge X_5, X_5 = X_1 \vee X_4$

- $\mathrm{Cap}_2(G) \geqslant \log_2(5) = 2.32...$

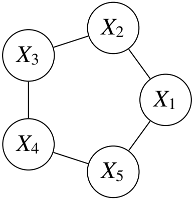
- In fact $\mathrm{Cap}_2(G) = \log_2(5) = 2.32...$

LRC codes on graphs - Example



- $\mathcal{C} = \{(0,0,0,0,0),(0,1,1,0,0),(0,0,0,1,1),(1,1,0,1,1),(1,1,1,0,1)\}$
- $X_1 = X_2 \wedge X_5, X_2 = X_1 \vee X_3, X_3 = X_2 \wedge \overline{X_4}, X_4 = \overline{X_3} \wedge X_5, X_5 = X_1 \vee X_4$
- $\mathrm{Cap}_2(G) \geqslant \log_2(5) = 2.32...$
- In fact $\mathrm{Cap}_2(G) = \log_2(5) = 2.32...$
- However $\mathrm{Cap}(G) = \mathrm{Cap}_4(G)$

LRC codes on graphs - Example



- $\mathcal{C} = \{(0,0,0,0,0),(0,1,1,0,0),(0,0,0,1,1),(1,1,0,1,1),(1,1,1,0,1)\}$
- $X_1 = X_2 \wedge X_5, X_2 = X_1 \vee X_3, X_3 = X_2 \wedge \overline{X_4}, X_4 = \overline{X_3} \wedge X_5, X_5 = X_1 \vee X_4$
- $\text{Cap}_2(G) \geqslant \log_2(5) = 2.32...$
- In fact $\text{Cap}_2(G) = \log_2(5) = 2.32...$
- However $\text{Cap}(G) = \text{Cap}_4(G) = 2.5$ [Blasiak, Kleinberg, Lubetzky 13, Christofides, Markstrom 11]

Storage Capacity

Generally

$$|\text{Maximum matching}| \leqslant \text{Cap}(G) \leqslant |\text{Vertex cover}|$$

The bounds are separated by a factor of 2.

For planar graphs there is a 1.5 approximation

• A. MAZUMDAR ET AL.*Storage capacity as an information-theoretic vertex cover and the index coding rate*, T-IT 2019

Results for planar graphs and cycles with chords

Duality between Storage Capacity and Index Coding

Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

Let $G = (V, E), |V| = n$, then

Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

Let $G=(V,E), |V|=n$, then

$$\mathrm{Cap}(G)+\mathrm{Index}(G)=n$$

Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

Let $G = (V, E)$, $|V| = n$, then

$$\mathrm{Cap}(G) + \mathrm{Index}(G) = n$$

Observations:

- Storage Capacity and Index coding are dual problems

Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

Let $G = (V, E), |V| = n$, then

$$\mathrm{Cap}(G) + \mathrm{Index}(G) = n$$

Observations:

- Storage Capacity and Index coding are dual problems
- upper bound on $\mathrm{Index}(G) \Rightarrow$ lower bound on $\mathrm{Cap}(G)$

Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

Let $G = (V, E), |V| = n$, then

$$\text{Cap}(G) + \text{Index}(G) = n$$

Observations:

- Storage Capacity and Index coding are dual problems
- upper bound on $\text{Index}(G) \Rightarrow$ lower bound on $\text{Cap}(G)$
- lower bound on $\text{Index}(G) \Rightarrow$ upper bound on $\text{Cap}(G)$

Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

Let $G = (V, E), |V| = n$, then

$$\text{Cap}(G) + \text{Index}(G) = n$$

Observations:

- Storage Capacity and Index coding are dual problems
- upper bound on $\text{Index}(G) \Rightarrow$ lower bound on $\text{Cap}(G)$
- lower bound on $\text{Index}(G) \Rightarrow$ upper bound on $\text{Cap}(G)$

Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

Let $G = (V, E), |V| = n$, then

$$\text{Cap}(G) + \text{Index}(G) = n$$

Recent works

• A. Mazumdar et al. *Storage capacity as an information-theoretic vertex cover and the index coding rate*, T-IT 2019
Approximated index coding capacity for planar graphs; found exactly for cycles with chords

• A. Golovnev, O. Regev, and O. Weinstein, *The minrank of random graphs*, T-IT 2019

Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

Let $G = (V, E), |V| = n$, then

$$\mathrm{Cap}(G) + \mathrm{Index}(G) = n$$

Recent works

• A. Mazumdar et al. *Storage capacity as an information-theoretic vertex cover and the index coding rate*, T-IT 2019
Approximated index coding capacity for planar graphs; found exactly for cycles with chords

• A. Golovnev, O. Regev, and O. Weinstein, *The minrank of random graphs*, T-IT 2019

Many open problems