Theory and Practice of Codes with Locality

Introduction

July 10, 2016
Information Era

• We live in Information Era
• Big Data players: Facebook, Google, MSFT, Amazon, Alibaba, Dropbox, etc.
• Node failures are the norm

Cluster of machines running Hadoop at Yahoo!
Information Era

- We live in Information Era
We live in Information Era

**Big Data players:** Facebook, Google, MSFT, Amazon, Alibaba, Dropbox, etc.
Information Era

- We live in Information Era

- **Big Data players:** Facebook, Google, MSFT, Amazon, Alibaba, Dropbox, etc.
Information Era

- We live in Information Era
- **Big Data players:** Facebook, Google, MSFT, Amazon, Alibaba, Dropbox, etc.
- Node failures are the norm
State-of-the-Art Coding technique

RAID: Redundant Array of Independent Disks

RAID 1 – Replication (typically 3x)

• Simple implementation!
• High availability of the information
• Can tolerate any 2 disk failures
• Widely used in Hadoop and many other systems
• Storage overhead of 200%!!!!!

RAID 6

• MDS codes with two parities $[n, k]$ MDS codes
• Can tolerate any $n-k$ disk failures
• FB uses $(14, 10)$ RS codes
• Poor handling of single disk failures (The Repair Problem)
State-of-the-Art Coding technique

RAID: Redundant Array of Independent Disks

- Simple implementation!
- High availability of the information
- Can tolerate any 2 disk failures
- Widely used in Hadoop and many other systems
- Storage overhead of 200%!!!

RAID 6
- MDS codes with two parities $[n,k]$ MDS codes
- Can tolerate any $n-k$ disk failures
- FB uses $\begin{pmatrix} 14 \\ 10 \end{pmatrix}$ RS codes
- Poor handling of single disk failures (The Repair Problem)
State-of-the-Art Coding technique

RAID: Redundant Array of Independent Disks

RAID 1 – Replication (typically 3x)
State-of-the-Art Coding technique

**RAID: Redundant Array of Independent Disks**

RAID 1 – Replication (typically 3x)
- Simple implementation!
- High availability of the information
- Can tolerate any 2 disk failures
- Widely used in *Hadoop* and many other systems
- Storage overhead of 200%!!!!!
State-of-the-Art Coding technique

RAID: Redundant Array of Independent Disks

RAID 1 – Replication (typically 3x)
- Simple implementation!
- High availability of the information
- Can tolerate any 2 disk failures
- Widely used in *Hadoop* and many other systems
- **Storage overhead of 200%!!!!!**

RAID 6
RAID: Redundant Array of Independent Disks

**RAID 1 – Replication (typically 3x)**
- Simple implementation!
- High availability of the information
- Can tolerate any 2 disk failures
- Widely used in *Hadoop* and many other systems
- Storage overhead of 200%!!!!!

**RAID 6**
- MDS codes with two parities
State-of-the-Art Coding technique

RAID: Redundant Array of Independent Disks

RAID 1 – Replication (typically 3x)
- Simple implementation!
- High availability of the information
- Can tolerate any 2 disk failures
- Widely used in Hadoop and many other systems
- Storage overhead of 200%!!!!!

RAID 6
- MDS codes with two parities

$[n, k]$ MDS codes
RAID: Redundant Array of Independent Disks

RAID 1 – Replication (typically 3x)
- Simple implementation!
- High availability of the information
- Can tolerate any 2 disk failures
- Widely used in Hadoop and many other systems
- Storage overhead of 200%!!!!!

RAID 6
- MDS codes with two parities

$[n, k]$ MDS codes
- Can tolerate any $n - k$ disk failures
State-of-the-Art Coding technique

RAID: Redundant Array of Independent Disks

RAID 1 – Replication (typically 3x)
- Simple implementation!
- High availability of the information
- Can tolerate any 2 disk failures
- Widely used in Hadoop and many other systems
- Storage overhead of 200%!!!!!

RAID 6
- MDS codes with two parities

\([n, k]\) MDS codes
- Can tolerate any \(n - k\) disk failures
- FB uses \((14, 10)\) RS codes
State-of-the-Art Coding technique

RAID: Redundant Array of Independent Disks

RAID 1 – Replication (typically 3x)
- Simple implementation!
- High availability of the information
- Can tolerate any 2 disk failures
- Widely used in Hadoop and many other systems
- Storage overhead of 200%!!!!!

RAID 6
- MDS codes with two parities

\([n, k]\) MDS codes
- Can tolerate any \(n - k\) disk failures
- FB uses \((14, 10)\) RS codes
- Poor handling of single disk failures (The Repair Problem)
Limitations of Reed-Solomon codes

Example:

\[
\begin{bmatrix}
14 \\
10
\end{bmatrix}
\]

RS code

• A disk is lost - Repair job starts to repair it
• Transmit information from 10 disks to recover one lost disk
• Generates 10x more traffic compared to replication for recovery of one disk
• If a large portion the data is RS-coded \( \Rightarrow \) saturation of the network
• Goal: Construct efficient codes with "good" repair process
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

- A disk is lost - Repair job starts to repair it
Limitations of Reed-Solomon codes

Example: [14, 10] RS code

- A disk is lost - Repair job starts to repair it
- Transmit information from 10 disks to recover one lost disk
Limitations of Reed-Solomon codes

Example: [14, 10] RS code

- A disk is lost - Repair job starts to repair it
- Transmit information from 10 disks to recover one lost disk
- Generates 10x more traffic compared to replication for recovery of one disk
Limitations of Reed-Solomon codes

Example: [14, 10] RS code

- A disk is lost - Repair job starts to repair it
- Transmit information from 10 disks to recover one lost disk
- Generates 10x more traffic compared to replication for recovery of one disk
- If a large portion the data is RS-coded $\implies$ saturation of the network
Limitations of Reed-Solomon codes

Example: [14, 10] RS code

- A disk is lost - Repair job starts to repair it
- Transmit information from 10 disks to recover one lost disk
- Generates 10x more traffic compared to replication for recovery of one disk
- If a large portion the data is RS-coded $\implies$ saturation of the network
- **Goal**: Construct efficient codes with "good" repair process
How “good” is the repair process?
How “good” is the repair process?

- Total bandwidth during the repair
How “good” is the repair process?

- Total bandwidth during the repair
  - Regenerating Codes - Seminal paper by Dimakis, Godfrey, Wu, Wainwright and Ramchandran
How “good” is the repair process?

- Total bandwidth during the repair
  - Regenerating Codes - Seminal paper by Dimakis, Godfrey, Wu, Wainwright and Ramchandran
  - Model allows connecting “many” nodes
How “good” is the repair process?

- Total bandwidth during the repair
  - Regenerating Codes - Seminal paper by Dimakis, Godfrey, Wu, Wainwright and Ramchandran
  - Model allows connecting “many” nodes
- Total number of disks participating in the repair
How “good” is the repair process?

- Total bandwidth during the repair
  - Regenerating Codes - Seminal paper by Dimakis, Godfrey, Wu, Wainwright and Ramchandran
  - Model allows connecting “many” nodes
- Total number of disks participating in the repair
  - The struggler
How “good” is the repair process?

- Total bandwidth during the repair
  - Regenerating Codes - Seminal paper by Dimakis, Godfrey, Wu, Wainwright and Ramchandran
  - Model allows connecting “many” nodes
- Total number of disks participating in the repair
  - The struggler
  - Locally Recoverable (LRC) Codes
Codes with locality: Plan

• LRC codes - Basics (Tamo)
• LRC codes - Constructions (Barg)
• LRC codes in practice (Barg)
• Break
• LRC codes - Bounds (Barg)
• Maximally Recoverable Codes (Tamo)
• LRC codes on graphs (Tamo)
Codes with locality: Plan

- LRC codes - Basics (Tamo)
Codes with locality: Plan

- LRC codes - Basics (Tamo)
- LRC codes - Constructions (Barg)
Codes with locality: Plan

- LRC codes - Basics (Tamo)
- LRC codes - Constructions (Barg)
- LRC codes in practice (Barg)
Codes with locality: Plan

- LRC codes - Basics (Tamo)
- LRC codes - Constructions (Barg)
- LRC codes in practice (Barg)
- Break 😊
Codes with locality: Plan

• LRC codes - Basics (Tamo)

• LRC codes - Constructions (Barg)

• LRC codes in practice (Barg)

• Break 😊

• LRC codes - Bounds (Barg)
Codes with locality: Plan

- LRC codes - Basics (Tamo)
- LRC codes - Constructions (Barg)
- LRC codes in practice (Barg)
- Break 😊
- LRC codes - Bounds (Barg)
- Maximally Recoverable Codes (Tamo)
Codes with locality: Plan

- LRC codes - Basics (Tamo)
- LRC codes - Constructions (Barg)
- LRC codes in practice (Barg)
- Break 😊
- LRC codes - Bounds (Barg)
- Maximally Recoverable Codes (Tamo)
- LRC codes on graphs (Tamo)
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

\[1 \quad \ldots \quad k - 1 \quad k\]
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

\[ 1 \ldots (k-1) \times k \quad k+1 \quad k+2 \ldots n \]
Locally Recoverable Codes - Definition

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

- Takes \(k\) blocks (symbols) \(\rightarrow\) produces \(n\) blocks
- An erasure has occurred
- Any symbol \(i\) has a recovery set \(R_i\) 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
- Any symbol \(i\) has a recovery set \(R_i\) 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
- Any symbol \(i\) has a recovery set \(R_i\) of \(r\) other symbols, \(r \ll k\)
Locally Recoverable Codes - Definition

\[(n, k, r)\] LRC Code

- Takes \(k\) blocks (symbols) → produces \(n\) blocks
- An erasure has occurred
- Any symbol \(i\) has a recovery set \(R_i\) of \(r\) other symbols, \(r \ll k\)

\[
\begin{array}{cccccc}
1 & \ldots & k-1 & \times & k+1 & k+2 \\
\text{\ldots} & \text{\ldots} & \text{\ldots} & \text{\ldots} & \text{\ldots} & \text{\ldots} \\
\end{array}
\]

\(r\) recovery set
Locally Recoverable Codes - Definition

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

- Takes \(k\) blocks (symbols) \(\rightarrow\) produces \(n\) blocks
- An erasure has occurred
- Any symbol \(i\) has a recovery set \(R_i\) of \(r\) other symbols, \(r \ll k\)
- Clearly \(1 \leq r \leq k\)
Early references on LRC codes

• J. Han and L. Lastras-Montano, ISIT 2007;
• C. Huang, M. Chen, and J. Li, Symp. Networks App. 2007;
• F. Oggier and A. Datta, 2013 - Survey on codes for distributed storage systems;
Early references on LRC codes

- J. Han and L. Lastras-Montano, *ISIT* 2007;
Early references on LRC codes

- J. Han and L. Lastras-Montano, *ISIT* 2007;

Early references on LRC codes

- J. Han and L. Lastras-Montano, *ISIT* 2007;


- F. Oggier and A. Datta 2013 - Survey on codes for distributed storage systems;
Early references on LRC codes

• J. Han and L. Lastras-Montano, ISIT 2007;

• C. Huang, M. Chen, and J. Li, Symp. Networks App. 2007;

• F. Oggier and A. Datta 2013 - Survey on codes for distributed storage systems;

Parameters of LRC codes

Let $C$ be an $(n, k, r)$ LRC code.

Assume $r | k$ and $r + 1 | n$. 

Theory and Practice of Codes with Locality July 10, 2016 9 / 22
Parameters of LRC codes

Let $C$ 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 $C$ be an $(n, k, r)$ LRC code

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

- Rate?
Parameters of LRC codes

Let $C$ be an $(n, k, r)$ LRC code

- Assume $r | k$ and $r + 1 | n$
- Rate?
- Minimum distance?
Parameters of LRC codes

Let $C$ be an $(n, k, r)$ LRC code

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

\[
\frac{k}{n} \leq \frac{r}{r + 1}.
\]
Parameters of LRC codes

Let $C$ be an $(n, k, r)$ LRC code

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

- The rate is bounded by

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

Proof:
Parameters of LRC codes

Let $C$ be an $(n, k, r)$ LRC code

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

- The rate is bounded by

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

Proof:

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

Let $C$ be an $(n, k, r)$ LRC code

- Assume $r | k$ and $r + 1 | n$
- The rate is bounded by
  \[
  \frac{k}{n} \leq \frac{r}{r + 1}.
  \]

Proof:

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

Let \( C \) be an \((n, k, r)\) LRC code

- Assume \( r \mid k \) and \( r + 1 \mid n \)
- The rate is bounded by
  \[
  \frac{k}{n} \leq \frac{r}{r + 1}.
  \]

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 recovery set \( \mathcal{R}_i \), \( |\mathcal{R}_i| \leq r \)
Parameters of LRC codes

Let $C$ be an $(n, k, r)$ LRC code

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

- The rate is bounded by
  \[
  \frac{k}{n} \leq \frac{r}{r + 1}.
  \]

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 recovery set $\mathcal{R}_i$, $|\mathcal{R}_i| \leq r$
  2. Free: the value of the $i$-th coordinate
Let $C$ be an $(n, k, r)$ LRC code

- Assume $r | k$ and $r + 1 | n$
- The rate is bounded by
  \[ \frac{k}{n} \leq \frac{r}{r + 1}. \]

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 recovery set $R_i$, $|R_i| \leq r$
  2. Free: the value of the $i$-th coordinate
  3. By exposing at most $\frac{nr}{r+1}$ coordinates, the exact codeword is determined
Parameters of LRC codes

Let $C$ be an $(n, k, r)$ LRC code

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

- The rate is bounded by

$$\frac{k}{n} \leq \frac{r}{r + 1}.$$
Parameters of LRC codes

Let $C$ be an $(n, k, r)$ LRC code

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

- The rate is bounded by

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

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

Let $C$ be an $(n, k, r)$ LRC code

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

- The rate is bounded by
  \[
  \frac{k}{n} \leq \frac{r}{r + 1}.
  \]

- The bound is tight (even over $\mathbb{F}_2$)
  - Partition the $k$ bits into $k/r$ sets of size $r$
Let $C$ be an $(n, k, r)$ LRC code

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

- The rate is bounded by

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

- 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 $C$ be an $(n, k, r)$ LRC code

- Assume $r|k$ and $r + 1|n$
Parameters of LRC codes

Let $C$ be an $(n, k, r)$ LRC code

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

- The minimum distance is bounded by

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

[GoPalan, Huang, Simitci, Yekhanin 12]
[Papailiopoulos, Dimakis 12]
Parameters of LRC codes

Let $C$ be an $(n, k, r)$ LRC code

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

- The minimum distance is bounded by

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

[Gopalan, Huang, Simitci, Yekhanin 12]
[Papailiopoulos, Dimakis 12]

Remarks:

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

Let $C$ be an $(n, k, r)$ LRC code

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

$$d \leq n - k - \left\lfloor \frac{k}{r} \right\rfloor + 2$$

[GoPalan, Huang, Simitci, Yekhanin 12]
[Papailiopoulos, Dimakis 12]

Remarks:

- Smaller locality $\implies$ lower failure resilience
Parameters of LRC codes

Let $C$ be an $(n, k, r)$ LRC code

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

- The minimum distance is bounded by

$$d \leq n - k - \left\lfloor \frac{k}{r} \right\rfloor + 2$$

[Gopalan, Huang, Simitci, Yekhanin 12]
[Papailiopoulos, Dimakis 12]

Remarks:

- Smaller locality $\implies$ lower failure resilience

- Generalization of the Singleton bound ($r = k$)
Parameters of LRC codes

Let $C$ be an $(n, k, r)$ LRC code

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

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

[Go palan, Huang, Simitci, Yekhanin 12]
[ Papai liopoulos, Dimakis 12]

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
Constructing optimal LRC codes

A carefully constructed random generating matrix gives an optimal LRC code. The optimal code is defined as:

\[(r+1) \lceil k \frac{r}{k} \rceil, k, r\]

This result was presented by Prasanth, Kamath, Lalitha, and Kumar in 2012.

Explicit constructions were proposed by Rawat, Koyluoglu, Silberstein, Vishwanath in 2014, and Gopalan, Huang, Jenkins, Yekhanin in 2014, Tamo, Papailiopoulos, Dimakis in 2014.
Constructing optimal LRC codes

- A carefully constructed random generating matrix gives an optimal LRC
Constructing optimal LRC codes

- A carefully constructed random generating matrix gives an optimal LRC
  1. Non explicit 😊
Constructing optimal LRC codes

• A carefully constructed random generating matrix gives an optimal LRC
  1. Non explicit ☹
  2. Field size is superpolynomial in the length ☹ (is it truly necessary?)

[Prasanth, Kamath, Lalitha, and Kumar 12]

[Rawat, Koyluoglu, Silberstein, Vishwanath 14, Gopalan, Huang, Jenkins, Yekhanin 14, Tamo, Papailiopoulos, Dimakis 14]
Constructing optimal LRC codes

- A carefully constructed random generating matrix gives an optimal LRC
  
  1. Non explicit ☹

  2. Field size is superpolynomial in the length ☹ (is it truly necessary?)

- Optimal \(( (r + 1)\left\lceil \frac{k}{r} \right\rceil, k, r \) \) LRC code [Prasanth, Kamath, Lalitha, and Kumar 12]
Constructing optimal LRC codes

- A carefully constructed random generating matrix gives an optimal LRC
  1. Non explicit ☺
  2. Field size is superpolynomial in the length ☹ (is it truly necessary?)

- Optimal \(((r + 1) \lceil \frac{k}{r} \rceil, k, r)\) LRC code [Prasanth, Kamath, Lalitha, and Kumar 12]

- Explicit constructions [Rawat, Koyluoglu, Silberstein, Vishwanath 14, Gopalan, Huang, Jenkins, Yekhanin 14, Tamo, Papailiopoulos, Dimakis 14]
Constructing optimal LRC codes

- A carefully constructed random generating matrix gives an optimal LRC
  1. Non explicit 😐
  2. Field size is superpolynomial in the length 😐 (is it truly necessary?)

- Optimal \(((r + 1)^\lceil \frac{k}{r} \rceil, k, r)\) LRC code [Prasanth, Kamath, Lalitha, and Kumar 12]

- Explicit constructions [Rawat, Koyluoglu, Silberstein, Vishwanath 14, Gopalan, Huang, Jenkins, Yekhanin 14, Tamo, Papailiopoulos, Dimakis 14]
  1. Any \(n, k, r\)
Constructing optimal LRC codes

- A carefully constructed random generating matrix gives an optimal LRC
  1. Non explicit ⊗
  2. Field size is superpolynomial in the length ⊗ (is it truly necessary?)

- Optimal \( ((r + 1)\lceil \frac{k}{r} \rceil, k, r) \) LRC code [Prasanth, Kamath, Lalitha, and Kumar 12]

- Explicit constructions [Rawat, Koyluoglu, Silberstein, Vishwanath 14, Gopalan, Huang, Jenkins, Yekhanin 14, Tamo, Papailiopoulos, Dimakis 14]
  1. Any \( n, k, r \)
  2. Field size is superpolynomial
Optimal LRC codes - Easy cases

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

2. An $(n, k)$ RS is an $(n, k, k)$ optimal LRC code.

3. $|F| = O(n)$

4. $r = 1$

1. $d \leq 2(n - k + 1)$

2. Duplication of an $(n/2, k)$ RS is an $(n, k, 1)$ optimal LRC code.

3. $|F| = O(n)$

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

Q: Generalize the optimal codes for $r = 1, k$ to codes with arbitrary $r$?
Optimal LRC codes - Easy cases

- $r = k$

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

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

Q: Generalize the optimal codes for $r = 1, k$ to codes with arbitrary $r$?
Optimal LRC codes - Easy cases

• $r = k$
  
  1. $d \leq n - k + 1$
Optimal LRC codes - Easy cases

- $r = k$
  1. $d \leq n - k + 1$
  2. An $(n, k)$ RS is an $(n, k, k)$ optimal LRC code
Optimal LRC codes - Easy cases

• \( r = k \)
  1. \( d \leq n - k + 1 \)
  2. An \((n, k)\) RS is an \((n, k, k)\) optimal LRC code
  3. \(|F| = O(n)|

Q: What happens for \(1 < r < k\)?
Q: Generalize the optimal codes for \(r = 1, k\) to codes with arbitrary \(r\)?
Optimal LRC codes - Easy cases

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

- $r = 1$

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

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

- $r = 1$
  1. $d \leq 2(\frac{n}{2} - k + 1)$

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

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

- $r = 1$
  1. $d \leq 2\left(\frac{n}{2} - k + 1\right)$
  2. Duplication of an $(n/2, k)$ RS is an $(n, k, 1)$ optimal LRC code
Optimal LRC codes - Easy cases

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

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

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

- \( r = 1 \)
  1. \( d \leq 2(\frac{n}{2} - k + 1) \)
  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 \( r \)?
Intuition behind the LRC construction
Intuition behind the LRC construction

\[(a_0, a_1, a_2, a_3)\]
Intuition behind the LRC construction

\[(a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3\]
Intuition behind the LRC construction

\[(a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3\]
Intuition behind the LRC construction

\[(a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1x + a_2x^2 + a_3x^3\]
Intuition behind the LRC construction

\((a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1x + a_2x^2 + a_3x^3\)
Intuition behind the LRC construction

\[(a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1x + a_2x^2 + a_3x^3\]
Intuition behind the LRC construction

\((a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3\)
Intuition behind the LRC construction

\[(a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1x + a_2x^2 + a_3x^3\]
Intuition behind the LRC construction

\((a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1x + a_2x^2 + a_3x^3\)
Intuition behind the LRC construction

\[(a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1x + a_2x^2 + a_3x^3\]
Intuition behind the LRC construction

\((a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \rightarrow f(x) = (f(x_1), f(x_2), \ldots, f(x_n))\)
Intuition behind the LRC construction

\[(a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \rightarrow f(x) = (f(x_1), f(x_2), \ldots, f(x_n))\]
Intuition behind the LRC construction

\[(a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \rightarrow f(x) = (f(x_1), f(x_2), \ldots, f(x_n))\]
Intuition behind the LRC construction

\((a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \rightarrow f(x) = (f(x_1), f(x_2), \ldots, f(x_n))\)
Intuition behind the LRC construction

\[(a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \rightarrow f(x) = (f(x_1), f(x_2), \ldots, f(x_n))\]
Intuition behind the LRC construction

\[(a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \rightarrow f(x) = (f(x_1), f(x_2), \ldots, f(x_n))\]
Intuition behind the LRC construction

$$(a_0, a_1, a_2, a_3) \rightarrow f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \rightarrow f(x) = (f(x_1), f(x_2), \ldots, f(x_n))$$

Only two points suffice to recover the lost point
Optimal \( (n, k, r) \) LRC code construction

Ingredients:
1. \( R_1, \ldots, R_n \) are disjoint subsets of the field \( F \), s.t. \( |R_i| = r + 1 \)
2. \( g(x) \in F[x] \) is a polynomial s.t.
   2.1 \( \deg(g(x)) = r + 1 \)
   2.2 \( g(x) \) is constant on each subset \( R_i \):
      \( g(\alpha) = g(\beta) \) for \( \alpha, \beta \in R_i \)

Encoding:
Given \( k \) information symbols \( a_{ij}, i = 0, \ldots, r-1, j = 0, \ldots, kr-1 \)
1. Define the polynomial \( f(x) = \sum_{i=0}^{r-1} x^i k \sum_{j=0}^{kr-1} a_{ij} g(x)^j \)
2. Store the length \( -n \) vector \( (f(\alpha) : \alpha \in \bigcup_i R_i) \)

Theorem:
This is an optimal \( (n, k, r) \) LRC code over \( F \).

Optimal \((n, k, r)\) LRC code construction

Ingredients:

Optimal \((n, k, r)\) LRC code construction

**Ingredients:**

1. \(\mathcal{R}_1, \ldots, \mathcal{R}_{\frac{n}{r+1}}\) are disjoint subsets of the field \(\mathbb{F}\), s.t. \(|\mathcal{R}_i| = r + 1\)

Optimal \((n, k, r)\) LRC code construction

**Ingredients:**

1. \(\mathcal{R}_1, \ldots, \mathcal{R}_{n_{r+1}}\) are disjoint subsets of the field \(\mathbb{F}\), s.t. \(|\mathcal{R}_i| = r + 1\)

2. \(g(x) \in \mathbb{F}[x]\) is a polynomial s.t.

---

Optimal \((n, k, r)\) LRC code construction

**Ingredients:**

1. \(\mathcal{R}_1, \ldots, \mathcal{R}_{\frac{n}{r+1}}\) are disjoint subsets of the field \(\mathbb{F}\), s.t. \(|\mathcal{R}_i| = r + 1\)

2. \(g(x) \in \mathbb{F}[x]\) is a polynomial s.t.
   
   2.1 \(\deg(g(x)) = r + 1\)

Optimal \((n, k, r)\) LRC code construction

**Ingredients:**

1. \(R_1, \ldots, R_{\frac{n}{r+1}}\) are disjoint subsets of the field \(\mathbb{F}\), s.t. \(|R_i| = r + 1\)

2. \(g(x) \in \mathbb{F}[x]\) is a polynomial s.t.
   
   2.1 \(\deg(g(x)) = r + 1\)

   2.2 \(g(x)\) is constant on each subset \(R_i\): \(g(\alpha) = g(\beta)\) for \(\alpha, \beta \in R_i\)

---

Optimal \( (n, k, r) \) LRC code construction

**Ingredients:**

1. \( \mathcal{R}_1, \ldots, \mathcal{R}_{n_{r+1}} \) are disjoint subsets of the field \( \mathbb{F} \), s.t. \( |\mathcal{R}_i| = r + 1 \)

2. \( g(x) \in \mathbb{F}[x] \) is a polynomial s.t.
   
   2.1 \( \text{deg} (g(x)) = r + 1 \)
   
   2.2 \( g(x) \) is constant on each subset \( \mathcal{R}_i \): \( g(\alpha) = g(\beta) \) for \( \alpha, \beta \in \mathcal{R}_i \)

**Encoding:**

Optimal \((n, k, r)\) LRC code construction

**Ingredients:**

1. \(\mathcal{R}_1, ..., \mathcal{R}_{\frac{n}{r+1}}\) are disjoint subsets of the field \(\mathbb{F}\), s.t. \(|\mathcal{R}_i| = r + 1\)

2. \(g(x) \in \mathbb{F}[x]\) is a polynomial s.t.
   
   2.1 \(\deg(g(x)) = r + 1\)
   
   2.2 \(g(x)\) is constant on each subset \(\mathcal{R}_i\): \(g(\alpha) = g(\beta)\) for \(\alpha, \beta \in \mathcal{R}_i\)

**Encoding:** Given \(k\) information symbols \(a_{i,j}\), \(i = 0, ..., r - 1, j = 0, ..., \frac{k}{r} - 1\)

Optimal \((n, k, r)\) LRC code construction

**Ingredients:**

1. \(\mathcal{R}_1, \ldots, \mathcal{R}_{\frac{n}{r+1}}\) are disjoint subsets of the field \(\mathbb{F}\), s.t. \(|\mathcal{R}_i| = r + 1\)

2. \(g(x) \in \mathbb{F}[x]\) is a polynomial s.t.
   
   2.1 \(\text{deg}(g(x)) = r + 1\)
   
   2.2 \(g(x)\) is constant on each subset \(\mathcal{R}_i\): \(g(\alpha) = g(\beta)\) for \(\alpha, \beta \in \mathcal{R}_i\)

**Encoding:** Given \(k\) information symbols \(a_{i,j}, i = 0, \ldots, r - 1, j = 0, \ldots, \frac{k}{r} - 1\)

1. Define the polynomial

\[
f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{\frac{k}{r}-1} a_{i,j} g(x)^j
\]

Optimal \((n, k, r)\) LRC code construction

**Ingredients:**

1. \(\mathcal{R}_1, \ldots, \mathcal{R}_{\frac{n}{r+1}}\) are disjoint subsets of the field \(\mathbb{F}\), s.t. \(|\mathcal{R}_i| = r + 1\)

2. \(g(x) \in \mathbb{F}[x]\) is a polynomial s.t.
   
   2.1 \(\deg(g(x)) = r + 1\)

   2.2 \(g(x)\) is constant on each subset \(\mathcal{R}_i\): \(g(\alpha) = g(\beta)\) for \(\alpha, \beta \in \mathcal{R}_i\)

**Encoding:** Given \(k\) information symbols \(a_{i,j}, i = 0, \ldots, r - 1, j = 0, \ldots, \frac{k}{r} - 1\)

1. Define the polynomial

\[
f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{\frac{k}{r}-1} a_{i,j} g(x)^j
\]

2. Store the length \(-n\) vector \((f(\alpha) : \alpha \in \cup_i \mathcal{R}_i)\)

Optimal \((n, k, r)\) LRC code construction

**Ingredients:**

1. \(\mathcal{R}_1, \ldots, \mathcal{R}_{\frac{n}{r+1}}\) are disjoint subsets of the field \(\mathbb{F}\), s.t. \(|\mathcal{R}_i| = r + 1\)

2. \(g(x) \in \mathbb{F}[x]\) is a polynomial s.t.
   
   2.1 \(\deg(g(x)) = r + 1\)
   
   2.2 \(g(x)\) is constant on each subset \(\mathcal{R}_i\): \(g(\alpha) = g(\beta)\) for \(\alpha, \beta \in \mathcal{R}_i\)

**Encoding:** Given \(k\) information symbols \(a_{i,j}, i = 0, \ldots, r - 1, j = 0, \ldots, \frac{k}{r} - 1\)

1. Define the polynomial

\[
f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{\frac{k}{r} - 1} a_{i,j} g(x)^j
\]

2. Store the length\(-n\) vector \((f(\alpha) : \alpha \in \cup_i \mathcal{R}_i)\)

**Theorem:** This is an optimal \((n, k, r)\) LRC code over \(\mathbb{F}\)

Optimal \((n, k, r)\) LRC code construction - Cont’d

\[ g(x) \text{ is constant on the sets } R_i, |R_i| = r + 1 \]

\[ f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j} g(x)^j \]

Locality:

Recover \(f(\alpha) = ?\) for \(\alpha \in R_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-1} a_{i,j} g(x)^j\)

2. \(f_i(x)\) is constant on the sets \(R_j\)

3. Define \(\delta(x) = \sum_{i=0}^{r-1} x^i f_i(R_1)\)

Claim:

\[ f(\beta) = \delta(\beta) \text{ for all } \beta \in R_1 \]

(In particular \(f(\alpha) = \delta(\alpha)\))

Proof:

\[ f(\beta) = \sum_{i=0}^{r-1} \beta^i f_i(\beta) = \sum_{i=0}^{r-1} \beta^i f_i(R_1) = \delta(\beta) \]

1. \(\deg(\delta(x)) \leq r - 1\)

2. \(r\) points on \(\delta(x)\) will suffice to recover \(\delta(x)\)

3. Read the \(r\) values \(
\{\delta(\beta) = f(\beta) : \beta \in R_1 \setminus \alpha\}\)
Optimal \((n,k,r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(R_i\), \(|R_i| = r + 1\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j}g(x)^j\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-r-1} a_{i,j}g(x)^j\)

**Locality:** Recover \(f(\alpha) =?\) for \(\alpha \in \mathcal{R}_1\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(R_i, |R_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-r-1} a_{i,j}g(x)^j\)

**Locality:** Recover \(f(\alpha) = ?\) for \(\alpha \in R_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-r-1} a_{i,j}g(x)^j\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(R_i, |R_i| = r + 1\)
- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j}g(x)^j = \sum_{i=0}^{r-1} x^i f_i(x)\)

**Locality:** Recover \(f(\alpha) =?\) for \(\alpha \in R_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-r-1} a_{i,j}g(x)^j\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j}g(x)^j = \sum_{i=0}^{r-1} x^i f_i(x)\)

**Locality:** Recover \(f(\alpha) =?\) for \(\alpha \in \mathcal{R}_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-1} a_{i,j}g(x)^j\)
2. \(f_i(x)\) is constant on the sets \(\mathcal{R}_j\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(R_i, |R_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j} g(x)^j = \sum_{i=0}^{r-1} x^i f_i(x)\)

**Locality:** Recover \(f(\alpha) =?\) for \(\alpha \in R_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-1} a_{i,j} g(x)^j\)
2. \(f_i(x)\) is constant on the sets \(R_j\)
3. Define \(\delta(x) = \sum_{i=0}^{r-1} x^i f_i(R_1)\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j}g(x)^j = \sum_{i=0}^{r-1} x^i f_i(x)\)

**Locality:** Recover \(f(\alpha) =?\) for \(\alpha \in \mathcal{R}_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-1} a_{i,j}g(x)^j\)
2. \(f_i(x)\) is constant on the sets \(\mathcal{R}_j\)
3. Define \(\delta(x) = \sum_{i=0}^{r-1} x^i f_i(\mathcal{R}_1)\)

**Claim:**
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-r-1} a_{i,j} g(x)^j = \sum_{i=0}^{r-1} x^i f_i(x)\)

**Locality:** Recover \(f(\alpha) = ?\) for \(\alpha \in \mathcal{R}_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-r-1} a_{i,j} g(x)^j\)
2. \(f_i(x)\) is constant on the sets \(\mathcal{R}_j\)
3. Define \(\delta(x) = \sum_{i=0}^{r-1} x^i f_i(\mathcal{R}_1)\)

**Claim:** \(f(\beta) = \delta(\beta)\) for all \(\beta \in \mathcal{R}_1\) (In particular \(f(\alpha) = \delta(\alpha)\))
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j} g(x)^j = \sum_{i=0}^{r-1} x^i f_i(x)\)

**Locality:** Recover \(f(\alpha) = \) for \(\alpha \in \mathcal{R}_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-1} a_{i,j} g(x)^j\)
2. \(f_i(x)\) is constant on the sets \(\mathcal{R}_j\)
3. Define \(\delta(x) = \sum_{i=0}^{r-1} x^i f_i(\mathcal{R}_1)\)

**Claim:** \(f(\beta) = \delta(\beta)\) for all \(\beta \in \mathcal{R}_1\) (In particular \(f(\alpha) = \delta(\alpha)\))

**Proof:**
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r+1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j} g(x)^j = \sum_{i=0}^{r-1} x^i f_i(x)\)

**Locality:** Recover \(f(\alpha) =?\) for \(\alpha \in \mathcal{R}_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-1} a_{i,j} g(x)^j\)
2. \(f_i(x)\) is constant on the sets \(\mathcal{R}_j\)
3. Define \(\delta(x) = \sum_{i=0}^{r-1} x^i f_i(\mathcal{R}_1)\)

**Claim:** \(f(\beta) = \delta(\beta)\) for all \(\beta \in \mathcal{R}_1\) (In particular \(f(\alpha) = \delta(\alpha))\)

**Proof:** \(f(\beta)\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j} g(x)^j = \sum_{i=0}^{r-1} x^i f_i(x)\)

**Locality:** Recover \(f(\alpha) = ?\) for \(\alpha \in \mathcal{R}_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-1} a_{i,j} g(x)^j\)
2. \(f_i(x)\) is constant on the sets \(\mathcal{R}_j\)
3. Define \(\delta(x) = \sum_{i=0}^{r-1} x^i f_i(\mathcal{R}_1)\)

**Claim:** \(f(\beta) = \delta(\beta)\) for all \(\beta \in \mathcal{R}_1\) (In particular \(f(\alpha) = \delta(\alpha)\))

**Proof:** \(f(\beta) = \sum_{i=0}^{r-1} \beta^i f_i(\beta)\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(R_i, |R_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{ij} g(x)^j = \sum_{i=0}^{r-1} x^i f_i(x)\)

**Locality:** Recover \(f(\alpha) = ?\) for \(\alpha \in R_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-1} a_{ij} g(x)^j\)
2. \(f_i(x)\) is constant on the sets \(R_j\)
3. Define \(\delta(x) = \sum_{i=0}^{r-1} x^i f_i(R_1)\)

**Claim:** \(f(\beta) = \delta(\beta)\) for all \(\beta \in R_1\) (In particular \(f(\alpha) = \delta(\alpha)\))

**Proof:** \(f(\beta) = \sum_{i=0}^{r-1} \beta^i f_i(\beta) = \sum_{i=0}^{r-1} \beta^i f_i(R_1)\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j} g(x)^j = \sum_{i=0}^{r-1} x^i f_i(x)\)

**Locality:** Recover \(f(\alpha) = ?\) for \(\alpha \in \mathcal{R}_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-1} a_{i,j} g(x)^j\)
2. \(f_i(x)\) is constant on the sets \(\mathcal{R}_j\)
3. Define \(\delta(x) = \sum_{i=0}^{r-1} x^i f_i(\mathcal{R}_1)\)

**Claim:** \(f(\beta) = \delta(\beta)\) for all \(\beta \in \mathcal{R}_1\) (In particular \(f(\alpha) = \delta(\alpha)\))

**Proof:** \(f(\beta) = \sum_{i=0}^{r-1} \beta^i f_i(\beta) = \sum_{i=0}^{r-1} \beta^i f_i(\mathcal{R}_1) = \delta(\beta)\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(R_i, |R_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j}g(x)^j = \sum_{i=0}^{r-1} x^i f_i(x)\)

**Locality:** Recover \(f(\alpha) = ?\) for \(\alpha \in R_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-1} a_{i,j}g(x)^j\)
2. \(f_i(x)\) is constant on the sets \(R_j\)
3. Define \(\delta(x) = \sum_{i=0}^{r-1} x^i f_i(R_1)\)

**Claim:** \(f(\beta) = \delta(\beta)\) for all \(\beta \in R_1\) (In particular \(f(\alpha) = \delta(\alpha)\))

**Proof:** \(f(\beta) = \sum_{i=0}^{r-1} \beta^i f_i(\beta) = \sum_{i=0}^{r-1} \beta^i f_i(R_1) = \delta(\beta)\)

1. \(\deg(\delta(x)) \leq r - 1\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j}g(x)^j = \sum_{i=0}^{r-1} x^i f_i(x)\)

**Locality:** Recover \(f(\alpha) = ?\) for \(\alpha \in \mathcal{R}_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-1} a_{i,j}g(x)^j\)

2. \(f_i(x)\) is constant on the sets \(\mathcal{R}_j\)

3. Define \(\delta(x) = \sum_{i=0}^{r-1} x^i f_i(\mathcal{R}_1)\)

**Claim:** \(f(\beta) = \delta(\beta)\) for all \(\beta \in \mathcal{R}_1\) (In particular \(f(\alpha) = \delta(\alpha)\))

**Proof:** \(f(\beta) = \sum_{i=0}^{r-1} \beta^i f_i(\beta) = \sum_{i=0}^{r-1} \beta^i f_i(\mathcal{R}_1) = \delta(\beta)\)

1. \(\deg(\delta(x)) \leq r - 1\)

2. \(r\) points on \(\delta(x)\) will suffice to recover \(\delta(x)\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j} g(x)^j = \sum_{i=0}^{r-1} x^i f_i(x)\)

**Locality:** Recover \(f(\alpha) = ?\) for \(\alpha \in \mathcal{R}_1\)

1. Define \(f_i(x) = \sum_{j=0}^{k-1} a_{i,j} g(x)^j\)
2. \(f_i(x)\) is constant on the sets \(\mathcal{R}_j\)
3. Define \(\delta(x) = \sum_{i=0}^{r-1} x^i f_i(\mathcal{R}_1)\)

**Claim:** \(f(\beta) = \delta(\beta)\) for all \(\beta \in \mathcal{R}_1\) (In particular \(f(\alpha) = \delta(\alpha)\))

**Proof:**

\[
f(\beta) = \sum_{i=0}^{r-1} \beta^i f_i(\beta) = \sum_{i=0}^{r-1} \beta^i f_i(\mathcal{R}_1) = \delta(\beta)
\]

1. \(\deg(\delta(x)) \leq r - 1\)
2. \(r\) points on \(\delta(x)\) will suffice to recover \(\delta(x)\)
3. Read the \(r\) values \(\{\delta(\beta) = f(\beta) : \beta \in \mathcal{R}_1 \setminus \alpha\}\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j} g(x)^j\)
• \( g(x) \) is constant on the sets \( \mathcal{R}_i, |\mathcal{R}_i| = r + 1 \)

• \( f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j} g(x)^j \)

**Minimum Distance:**
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j} g(x)^j\)

**Minimum Distance:**

1. Minimum distance = Minimum weight
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j} g(x)^j\)

Minimum Distance:

1. Minimum distance = Minimum weight

2. \(\text{deg}(f(x)) \leq r - 1 + (r + 1)(\frac{k}{r} - 1)\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j} g(x)^j\)

**Minimum Distance:**

1. Minimum distance = Minimum weight

2. \(\deg(f(x)) \leq r - 1 + (r + 1)(\frac{k}{r} - 1) = k + \frac{k}{r} - 2\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j}g(x)^j\)

Minimum Distance:

1. Minimum distance = Minimum weight

2. \(\deg(f(x)) \leq r - 1 + (r + 1)(\frac{k}{r} - 1) = k + \frac{k}{r} - 2\)

3. \(\implies d \geq n - (k + \frac{k}{r} - 2)\)
Optimal \((n, k, r)\) LRC code construction - Cont’d

- \(g(x)\) is constant on the sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\)

- \(f(x) = \sum_{i=0}^{r-1} x^i \sum_{j=0}^{k-1} a_{i,j} g(x)^j\)

**Minimum Distance:**

1. Minimum distance = Minimum weight

2. \(\deg(f(x)) \leq r - 1 + (r + 1)(\frac{k}{r} - 1) = k + \frac{k}{r} - 2\)

3. \(\implies d \geq n - (k + \frac{k}{r} - 2)\)

4. Equality follows since the code is an \((n, k, r)\) LRC code
Example: $(9, 4, 2)$ LRC over $\mathbb{F}_{13}$
Example: $(9, 4, 2)$ LRC over $\mathbb{F}_{13}$

- $\mathcal{R}_1 = \{1, 3, 9\}$, $\mathcal{R}_2 = \{2, 6, 5\}$, $\mathcal{R}_3 = \{4, 12, 10\}$
Example: \((9, 4, 2)\) LRC over \(\mathbb{F}_{13}\)

- \(\mathcal{R}_1 = \{1, 3, 9\}, \mathcal{R}_2 = \{2, 6, 5\}, \mathcal{R}_3 = \{4, 12, 10\}\)

- The polynomial \(g(x) = x^3\) is constant on each set \(\mathcal{R}_i\)
Example: (9, 4, 2) LRC over $\mathbb{F}_{13}$

- $\mathcal{R}_1 = \{1, 3, 9\}$, $\mathcal{R}_2 = \{2, 6, 5\}$, $\mathcal{R}_3 = \{4, 12, 10\}$
- The polynomial $g(x) = x^3$ is constant on each set $\mathcal{R}_i$
- $(a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) = (1, 1, 1, 1)$
Example: (9, 4, 2) LRC over $\mathbb{F}_{13}$

- $\mathcal{R}_1 = \{1, 3, 9\}$, $\mathcal{R}_2 = \{2, 6, 5\}$, $\mathcal{R}_3 = \{4, 12, 10\}$

- The polynomial $g(x) = x^3$ is constant on each set $\mathcal{R}_i$

- $(a_0, 0, a_1, 0, a_1, 1) = (1, 1, 1, 1) \rightarrow f(x) = \sum_{i=0}^{1} x^i \sum_{j=0}^{1} a_{i,j} (x^3)^j$
Example: $(9, 4, 2)$ LRC over $\mathbb{F}_{13}$

- $\mathcal{R}_1 = \{1, 3, 9\}, \mathcal{R}_2 = \{2, 6, 5\}, \mathcal{R}_3 = \{4, 12, 10\}$

- The polynomial $g(x) = x^3$ is constant on each set $\mathcal{R}_i$

- $(a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) = (1, 1, 1, 1) \rightarrow f(x) = \sum_{i=0}^{1} x^i \sum_{j=0}^{1} a_{i,j} (x^3)^j = 1 + x + x^3 + x^4$
Example: $(9, 4, 2)$ LRC over $\mathbb{F}_{13}$

- $\mathcal{R}_1 = \{1, 3, 9\}, \mathcal{R}_2 = \{2, 6, 5\}, \mathcal{R}_3 = \{4, 12, 10\}$

- The polynomial $g(x) = x^3$ is constant on each set $\mathcal{R}_i$

- $(a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) = (1, 1, 1, 1) \rightarrow f(x) = \sum_{i=0}^{1} x^i \sum_{j=0}^{1} a_{i,j} (x^3)^j = 1 + x + x^3 + x^4$

- Store the evaluation of $f(x)$ at $\cup_i \mathcal{R}_i$
Example: \((9, 4, 2)\) LRC over \(\mathbb{F}_{13}\)

- \(\mathcal{R}_1 = \{1, 3, 9\}, \mathcal{R}_2 = \{2, 6, 5\}, \mathcal{R}_3 = \{4, 12, 10\}\)

- The polynomial \(g(x) = x^3\) is constant on each set \(\mathcal{R}_i\)

- \((a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) = (1, 1, 1, 1) \rightarrow f(x) = \sum_{i=0}^{1} x^i \sum_{j=0}^{1} a_{i,j} (x^3)^j = 1 + x + x^3 + x^4\)

- Store the evaluation of \(f(x)\) at \(\bigcup_i \mathcal{R}_i\)

\[
(f(1), f(3), f(9), f(2), f(6), f(5), f(4), f(12), f(10)) = (4, 8, 7, 1, 11, 2, 0, 0, 0)
\]
Example: $(9, 4, 2)$ LRC over $\mathbb{F}_{13}$

- $\mathcal{R}_1 = \{1, 3, 9\}$, $\mathcal{R}_2 = \{2, 6, 5\}$, $\mathcal{R}_3 = \{4, 12, 10\}$

- The polynomial $g(x) = x^3$ is constant on each set $\mathcal{R}_i$

- $(a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) = (1, 1, 1, 1) \rightarrow f(x) = \sum_{i=0}^{1} x^i \sum_{j=0}^{1} a_{i,j} (x^3)^j = 1 + x + x^3 + x^4$

- Store the evaluation of $f(x)$ at $\bigcup_i \mathcal{R}_i$

$$f(1) = ?$$
Example: $(9, 4, 2)$ LRC over $\mathbb{F}_{13}$

- $\mathcal{R}_1 = \{1, 3, 9\}$, $\mathcal{R}_2 = \{2, 6, 5\}$, $\mathcal{R}_3 = \{4, 12, 10\}$

- The polynomial $g(x) = x^3$ is constant on each set $\mathcal{R}_i$

- $(a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) = (1, 1, 1, 1) \rightarrow f(x) = \sum_{i=0}^{1} x^i \sum_{j=0}^{1} a_{i,j} (x^3)^j = 1 + x + x^3 + x^4$

- Store the evaluation of $f(x)$ at $\bigcup_i \mathcal{R}_i$

\[
(f(0), f(3), f(9), f(2), f(6), f(5), f(4), f(12), f(10)) = (?, 8, 7, 1, 11, 2, 0, 0, 0)
\]

- $f(1) =$?

Local Recovery:
Example: $(9, 4, 2)$ LRC over $\mathbb{F}_{13}$

- $\mathcal{R}_1 = \{1, 3, 9\}, \mathcal{R}_2 = \{2, 6, 5\}, \mathcal{R}_3 = \{4, 12, 10\}$

- The polynomial $g(x) = x^3$ is constant on each set $\mathcal{R}_i$

- $(a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) = (1, 1, 1, 1) \rightarrow f(x) = \sum_{i=0}^1 x^i \sum_{j=0}^1 a_{i,j}(x^3)^j = 1 + x + x^3 + x^4$

- Store the evaluation of $f(x)$ at $\bigcup_i \mathcal{R}_i$

\[
(0, f(3), f(9), f(2), f(6), f(5), f(4), f(12), f(10)) = (\text{8,7,1,11,2,0,0,0})
\]

- $f(1) = ?$

  Local Recovery:

  - $\deg(\delta(x)) \leq 1$
Example: (9, 4, 2) LRC over $\mathbb{F}_{13}$

- $\mathcal{R}_1 = \{1, 3, 9\}$, $\mathcal{R}_2 = \{2, 6, 5\}$, $\mathcal{R}_3 = \{4, 12, 10\}$

- The polynomial $g(x) = x^3$ is constant on each set $\mathcal{R}_i$

- $(a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) = (1, 1, 1, 1) \rightarrow f(x) = \sum_{i=0}^{1} x^i \sum_{j=0}^{1} a_{i,j} (x^3)^j = 1 + x + x^3 + x^4$

- Store the evaluation of $f(x)$ at $\cup_i \mathcal{R}_i$

$$(f(0), f(3), f(9), f(2), f(6), f(5), f(4), f(12), f(10)) = (\times8,7,1,11,2,0,0,0)$$

$f(1) = ?$

Local Recovery:

- $\deg(\delta(x)) \leq 1 \quad f(3) = \delta(3) = 8$
Example: \((9, 4, 2)\) LRC over \(\mathbb{F}_{13}\)

- \(\mathcal{R}_1 = \{1, 3, 9\}, \mathcal{R}_2 = \{2, 6, 5\}, \mathcal{R}_3 = \{4, 12, 10\}\)

- The polynomial \(g(x) = x^3\) is constant on each set \(\mathcal{R}_i\)

- \((a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) = (1, 1, 1, 1) \rightarrow f(x) = \sum_{i=0}^{1} x^i \sum_{j=0}^{1} a_{i,j} (x^3)^j = 1 + x + x^3 + x^4\)

- Store the evaluation of \(f(x)\) at \(\bigcup_i \mathcal{R}_i\)

\[
(f(0), f(3), f(9), f(2), f(6), f(5), f(4), f(12), f(10)) = (8, 8, 7, 1, 11, 2, 0, 0, 0)
\]

- Local Recovery:
  - \(\deg(\delta(x)) \leq 1\)
  - \(f(3) = \delta(3) = 8\)
  - \(f(9) = \delta(9) = 7\)
Example: $(9, 4, 2)$ LRC over $\mathbb{F}_{13}$

- $\mathcal{R}_1 = \{1, 3, 9\}$, $\mathcal{R}_2 = \{2, 6, 5\}$, $\mathcal{R}_3 = \{4, 12, 10\}$

- The polynomial $g(x) = x^3$ is constant on each set $\mathcal{R}_i$

- $(a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) = (1, 1, 1, 1) \rightarrow f(x) = \sum_{i=0}^{1} x^i \sum_{j=0}^{1} a_{i,j} (x^3)^j = 1 + x + x^3 + x^4$

- Store the evaluation of $f(x)$ at $\cup_i \mathcal{R}_i$

- $(\times, f(3), f(9), f(2), f(6), f(5), f(4), f(12), f(10)) = (\times, 8, 7, 1, 11, 2, 0, 0, 0)$

  $f(1) = ?$

  Local Recovery:

  - $\deg(\delta(x)) \leq 1$
    - $f(3) = \delta(3) = 8$
    - $f(9) = \delta(9) = 7$

  - $\delta(x) = 2x + 2$
Example: $(9, 4, 2)$ LRC over $\mathbb{F}_{13}$

- $\mathcal{R}_1 = \{1, 3, 9\}, \mathcal{R}_2 = \{2, 6, 5\}, \mathcal{R}_3 = \{4, 12, 10\}$

- The polynomial $g(x) = x^3$ is constant on each set $\mathcal{R}_i$

- $(a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) = (1, 1, 1, 1) \rightarrow f(x) = \sum_{i=0}^{1} x^i \sum_{j=0}^{1} a_{i,j} (x^3)^j = 1 + x + x^3 + x^4$

- Store the evaluation of $f(x)$ at $\bigcup_i \mathcal{R}_i$

\[
(f_1, f_3, f_9, f_2, f_6, f_5, f_4, f_{12}, f_{10}) = (8, 7, 1, 11, 2, 0, 0, 0)
\]

$f(1) = ?$

Local Recovery:

- $\deg(\delta(x)) \leq 1$  \hspace{1cm} $f(3) = \delta(3) = 8$  \hspace{1cm} $f(9) = \delta(9) = 7$

- $\delta(x) = 2x + 2$  \hspace{1cm} $f(1) = \delta(1) = 4$
Example: \((9, 4, 2)\) LRC over \(\mathbb{F}_{13}\)

- \(\mathcal{R}_1 = \{1, 3, 9\}, \mathcal{R}_2 = \{2, 6, 5\}, \mathcal{R}_3 = \{4, 12, 10\}\)

- The polynomial \(g(x) = x^3\) is constant on each set \(\mathcal{R}_i\)

- \((a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}) = (1, 1, 1, 1) \rightarrow f(x) = \sum_{i=0}^{1} x^i \sum_{j=0}^{1} a_{i,j} (x^3)^j = 1 + x + x^3 + x^4\)

- Store the evaluation of \(f(x)\) at \(\bigcup_i \mathcal{R}_i\)

\[
(f(1), f(3), f(9), f(2), f(6), f(5), f(4), f(12), f(10)) = (4, 8, 7, 1, 11, 2, 0, 0, 0)
\]

- \(f(1) =?\)

**Local Recovery:**

- \(\text{deg}(\delta(x)) \leq 1\) \quad \(f(3) = \delta(3) = 8\) \quad \(f(9) = \delta(9) = 7\)

- \(\delta(x) = 2x + 2\) \quad \(f(1) = \delta(1) = 4\)
Ingredients:

1. $R_1, \ldots, R_n$ are disjoint subsets of the field $\mathbb{F}$, s.t. $|R_i| = r + 1$

2. A polynomial $g(x)$ of degree $r + 1$

3. $g(\alpha) = g(\beta)$ for $\alpha, \beta \in R_i$

Claim:

Let $H$ be a subgroup of $\mathbb{F}^*$ or $\mathbb{F}^+$, then the annihilator polynomial of $H$

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

is constant on each coset of $H$.
Optimal \((n, k, r)\) LRC code construction

**Ingredients:**

1. \(R_1, \ldots, R_n\) are disjoint subsets of the field \(F\), s.t. \(|R_i| = r + 1\)
2. A polynomial \(g(x)\) of degree \(r + 1\)
3. \(g(x)\) is constant on each subset \(R_i\): \(g(\alpha) = g(\beta)\) for \(\alpha, \beta \in R_i\)

**Claim:**

Let \(H\) be a subgroup of \(F^*\) or \(F^+\), then the annihilator polynomial of \(H\)

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

is constant on each coset of \(H\).
Optimal \((n, k, r)\) LRC code construction

Ingredients:

1. \(\mathcal{R}_1, \ldots, \mathcal{R}_{n_{r+1}}\) are disjoint subsets of the field \(\mathbb{F}\), s.t. \(|\mathcal{R}_i| = r + 1\)

2. A polynomial \(g(x)\) of degree \(r + 1\)

3. \(g(x)\) is constant on each subset \(\mathcal{R}_i\): \(g(\alpha) = g(\beta)\) for \(\alpha, \beta \in \mathcal{R}_i\)
Optimal \((n, k, r)\) LRC code construction

**Ingredients:**

1. \(\mathcal{R}_1, \ldots, \mathcal{R}_{\frac{n}{r+1}}\) are disjoint subsets of the field \(\mathbb{F}\), s.t. \(|\mathcal{R}_i| = r + 1\)

2. A polynomial \(g(x)\) of degree \(r + 1\)

3. \(g(x)\) is constant on each subset \(\mathcal{R}_i\): \(g(\alpha) = g(\beta)\) for \(\alpha, \beta \in \mathcal{R}_i\)

**Claim:**
Optimal \((n, k, r)\) LRC code construction

**Ingredients:**

1. \(R_1, \ldots, R_{\frac{n}{r+1}}\) are disjoint subsets of the field \(\mathbb{F}\), s.t. \(|R_i| = r + 1\)

2. A polynomial \(g(x)\) of degree \(r + 1\)

3. \(g(x)\) is constant on each subset \(R_i\): \(g(\alpha) = g(\beta)\) for \(\alpha, \beta \in R_i\)

**Claim:** Let \(H\) be a subgroup of \(\mathbb{F}^*\) or \(\mathbb{F}^+\), then the annihilator polynomial of \(H\)

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

is constant on each coset of \(H\)
Availability problem
Availability problem

“Hot data” accessed simultaneously by a very large number of users
“Hot data” accessed simultaneously by a very large number of users
Availability problem - Cont’d

Main advantage of replication - High availability for hot data

Goal:

A code with high availability and small overhead

Solution:

LRC code with multiple disjoint recovery sets (codes with availability)
Availability problem - Cont’d

- Main advantage of replication - High availability for hot data
Availability problem - Cont’d

- Main advantage of replication - High availability for hot data
- **Goal:** A code with high availability and small overhead
Availability problem - Cont’d

- Main advantage of replication - High availability for hot data
- **Goal:** A code with high availability and small overhead
- **Solution:** LRC code with multiple **disjoint** recovery sets (codes with availability)
Idea behind constructing codes with multiple recovery sets
Idea behind constructing codes with multiple recovery sets

Subspace of polynomials that provide locality $r_1$
Idea behind constructing codes with multiple recovery sets

Subspace of polynomials that provide locality $r_1$

Ex: The space of polynomials $\{a_0 + a_1x + a_3x^3 + a_4x^4: a_i \in F_{13}\}$
Idea behind constructing codes with multiple recovery sets

Subspace of polynomials that provide locality $r_2$

Subspace of polynomials that provide locality $r_1$

Ex: The space of polynomials $\{a_0 + a_1x + a_3x^3 + a_4x^4: a_i \in F_{13}\}$
Idea behind constructing codes with multiple recovery sets

Subspace of polynomials that provide locality $r_2$

Subspace of polynomials that provide locality $r_1$

Ex: The space of polynomials $\{a_0 + a_1 x + a_3 x^3 + a_4 x^4: a_i \in F_{13}\}$
LRC code with 2 recovery sets - Example

Example: \((12, 6, \{2, 3\})\) over \(\mathbb{F}_{13}\).

1. Encoding polynomial:

\[ \left( a_0, a_1, a_6, a_4, a_9, a_{10} \right) \mapsto f(x) = a_0 + a_1 x + a_4 x^4 + a_6 x^6 + a_9 x^9 + a_{10} x^{10} \]
LRC code with 2 recovery sets - Example

- Example: $(12, 6, \{2, 3\})$ over $\mathbb{F}_{13}$
LRC code with 2 recovery sets - Example

- **Example:** $(12, 6, \{2, 3\})$ over $\mathbb{F}_{13}$

  1. Encoding polynomial:

     $$ (a_0, a_1, a_6, a_4, a_9, a_{10}) \mapsto f(x) = a_0 + a_1x + a_4x^4 + a_6x^6 + a_9x^9 + a_{10}x^{10} $$
LRC code with 2 recovery sets - Example

- **Example:** \((12, 6, \{2, 3\})\) over \(\mathbb{F}_{13}\)

1. Encoding polynomial:

\[(a_0, a_1, a_6, a_4, a_9, a_{10}) \mapsto f(x) = a_0 + a_1 x + a_4 x^4 + a_6 x^6 + a_9 x^9 + a_{10} x^{10}\]

2. Evaluated points: \(\mathbb{F}_{13}^*\)
LRC code with 2 recovery sets - Example

- **Example:** \((12, 6, \{2, 3\})\) over \(\mathbb{F}_{13}\)

1. Encoding polynomial:
   \[
   (a_0, a_1, a_6, a_4, a_9, a_{10}) \mapsto f(x) = a_0 + a_1 x + a_4 x^4 + a_6 x^6 + a_9 x^9 + a_{10} x^{10}
   \]

2. Evaluated points: \(\mathbb{F}^*_{13}\)

\((18, 6, \{1,1\}) 3x\ Replication \quad (12, 6, \{2,3\})\) LRC
LRC code with 2 recovery sets - Example

• **Example:** $(12, 6, \{2, 3\})$ over $\mathbb{F}_{13}$

  1. Encoding polynomial:

     $$(a_0, a_1, a_6, a_4, a_9, a_{10}) \mapsto f(x) = a_0 + a_1 x + a_4 x^4 + a_6 x^6 + a_9 x^9 + a_{10} x^{10}$$

  2. Evaluated points: $\mathbb{F}_{13}^*$

$(18, 6, \{1, 1\})$ 3x Replication  $(12, 6, \{2, 3\})$ LRC

• Overhead of 200%
LRC code with 2 recovery sets - Example

- **Example:** \((12, 6, \{2, 3\})\) over \(\mathbb{F}_{13}\)

  1. Encoding polynomial:

\[
(a_0, a_1, a_6, a_4, a_9, a_{10}) \mapsto f(x) = a_0 + a_1 x + a_4 x^4 + a_6 x^6 + a_9 x^9 + a_{10} x^{10}
\]

  2. Evaluated points: \(\mathbb{F}^*_{13}\)

\((18, 6, \{1, 1\}) 3x Replication \quad (12, 6, \{2, 3\}) LRC\)

- Overhead of 200%
- Overhead of 100%
LRC code with 2 recovery sets - Example

- **Example:** $(12, 6, \{2, 3\})$ over $\mathbb{F}_{13}$

  1. Encoding polynomial:

     $$(a_0, a_1, a_6, a_4, a_9, a_{10}) \mapsto f(x) = a_0 + a_1 x + a_4 x^4 + a_6 x^6 + a_9 x^9 + a_{10} x^{10}$$

  2. Evaluated points: $\mathbb{F}^*_{13}$

$(18,6,\{1,1\})$ 3x Replication $(12,6,\{2,3\})$ LRC

- Overhead of 200%
- Can tolerate any 2 disk failures
- Overhead of 100%
LRC code with 2 recovery sets - Example

- **Example:** \((12, 6, \{2, 3\})\) over \(\mathbb{F}_{13}\)

  1. Encoding polynomial:

  \[(a_0, a_1, a_6, a_4, a_9, a_{10}) \mapsto f(x) = a_0 + a_1 x + a_4 x^4 + a_6 x^6 + a_9 x^9 + a_{10} x^{10}\]

  2. Evaluated points: \(\mathbb{F}_{13}^*\)

\[(18, 6, \{1, 1\}) \text{ 3x Replication} \quad (12, 6, \{2, 3\}) \text{ LRC}\]

- Overhead of 200%
- Can tolerate any 2 disk failures
- Overhead of 100%
- Can tolerate any 3 disk failures
LRC code with 2 recovery sets - Example

- **Example:** \((12, 6, \{2, 3\})\) over \(\mathbb{F}_{13}\)

1. Encoding polynomial:

\[
    (a_0, a_1, a_6, a_4, a_9, a_{10}) \mapsto f(x) = a_0 + a_1x + a_4x^4 + a_6x^6 + a_9x^9 + a_{10}x^{10}
\]

2. Evaluated points: \(\mathbb{F}^*_{13}\)

\((18,6,\{1,1\})\) \textit{3x Replication} \hspace{1cm} \((12,6,\{2,3\})\) \textit{LRC}

- Overhead of 200%
- Can tolerate any 2 disk failures
- Recovery sets \(\{\bullet, \bullet, \bullet\}\)

- Overhead of 100%
- Can tolerate any 3 disk failures
LRC code with 2 recovery sets - Example

- **Example:** \((12, 6, \{2, 3\})\) over \(\mathbb{F}_{13}\)

  1. Encoding polynomial:

\[
(a_0, a_1, a_6, a_4, a_9, a_{10}) \mapsto f(x) = a_0 + a_1 x + a_4 x^4 + a_6 x^6 + a_9 x^9 + a_{10} x^{10}
\]

  2. Evaluated points: \(\mathbb{F}_{13}^*\)

(\(18,6,\{1,1\}\)) 3x Replication    (\(12,6,\{2,3\}\)) LRC

- Overhead of 200%
- Can tolerate any 2 disk failures
- Recovery sets \(\{\bullet, \bullet, \bullet\}\)

- Overhead of 100%
- Can tolerate any 3 disk failures
- Recovery sets \(\{\bullet, \bullet, \bullet, \bullet, \bullet\}\)
Codes with locality: Plan

- LRC codes - Basics (Tamo)
- LRC codes - Constructions (Barg)
- LRC codes in practice (Barg)
- Break 😊
- LRC codes - Bounds (Barg)
- Maximally Recoverable Codes (Tamo)
- LRC codes on graphs (Tamo)
Constructions of LRC codes
From RS codes to other related families

In this part our goal is to construct new families of LRC codes
In this part our goal is to construct new families of LRC codes

**Advantages of the LRC RS construction**

- small alphabet $q \approx n$
- large distance
- easy processing
From RS codes to other related families

In this part our goal is to construct new families of LRC codes

Advantages of the LRC RS construction

- small alphabet $q \approx n$
- large distance
- easy processing

Limitations

- $n \leq q$
In this part our goal is to construct new families of LRC codes

**Advantages of the LRC RS construction**

- small alphabet $q \approx n$
- large distance
- easy processing

**Limitations**

- $n \leq q$

**Fixed $q$, large $n$?**
From RS codes to other related families

In this part our goal is to construct new families of LRC codes

Advantages of the LRC RS construction

- small alphabet \( q \approx n \)
- large distance
- easy processing

Limitations

- \( n \leq q \)

**Fixed** \( q \), large \( n \)?

Possibilities: **Reduce** \( q \) or **Increase** \( n \)
From RS codes to other related families

In this part our goal is to construct new families of LRC codes

Advantages of the LRC RS construction

- small alphabet $q \approx n$
- large distance
- easy processing

Limitations

- $n \leq q$

Fixed $q$, large $n$?

Possibilities: Reduce $q$ or Increase $n$

- Take a subfield subcode of an LRC RS code (e.g., a binary code)
  - The analysis is simplified if we take a cyclic LRC RS code
- Take a large subset of sampling points (points on an algebraic curve)
Cyclic LRC codes

Cyclic codes form a classic topic in coding theory: BCH codes, RM codes, many other well-studied code families are cyclic.

Cyclic $q$-ary LRC codes

Recall the LRC RS construction:

Data: subset of points $\mathcal{P} = (P_1, \ldots, P_n) \subset \mathbb{F}_q$; linear space of polynomials $V = \langle x^ib(x)^i \rangle$, $\dim V = k$, $b(x)$ constant on $(r + 1)$-subblocks of the set $\mathcal{P}$;

$$V \rightarrow \mathbb{C}$$

$$f_a \mapsto ev_{\mathcal{P}}(f_a) = (f_a(P_1), \ldots, f_a(P_n))$$
Cyclic LRC codes

Cyclic codes form a classic topic in coding theory: BCH codes, RM codes, many other well-studied code families are cyclic.

Cyclic $q$-ary LRC codes

Recall the LRC RS construction:

**Data:** subset of points $\mathcal{P} = (P_1, \ldots, P_n) \subset \mathbb{F}_q$; linear space of polynomials $V = \langle x^i b(x)^i \rangle$, $\dim V = k$, $b(x)$ constant on $(r + 1)$-subblocks of the set $\mathcal{P}$;

$$V \rightarrow \mathbb{C}$$

$$f_a \mapsto ev_{\mathcal{P}}(f_a) = (f_a(P_1), \ldots, f_a(P_n))$$

Additional assumptions: $n | (q - 1)$; $(r + 1) | n$; $r | k$

$$\mathcal{P} = \{1, \alpha, \ldots, \alpha^{n-1}\}; \quad V = \left\langle \sum_{i=0}^{(k \cdot \frac{(r+1)-2}{r})} a_i x^i, a_i \in \mathbb{F}_q \right\rangle$$

(\alpha = n\text{th root of unity})
Cyclic LRC codes

Cyclic codes form a classic topic in coding theory: BCH codes, RM codes, many other well-studied code families are cyclic.

Cyclic $q$-ary LRC codes

Recall the LRC RS construction:

**Data:** subset of points $\mathcal{P} = (P_1, \ldots, P_n) \subset \mathbb{F}_q$; linear space of polynomials $V = \langle x^i b(x)^i \rangle$, $\dim V = k$, $b(x)$ constant on $(r+1)$-subblocks of the set $\mathcal{P}$;

\[ V \rightarrow \mathbb{C} \]

\[ f_a \mapsto ev_{\mathcal{P}}(f_a) = (f_a(P_1), \ldots, f_a(P_n)) \]

Additional assumptions: $n | (q - 1); (r + 1)|n; r|k$

\[ \mathcal{P} = \{1, \alpha, \ldots, \alpha^{n-1}\}; \quad V = \left\langle \sum_{i=0}^{k \frac{r(r+1)-2}{r}} a_i x^i, a_i \in \mathbb{F}_q \right\rangle \quad (1) \]

($\alpha$ - $n$th root of unity)

We obtain a cyclic code $\mathbb{C}$:

\[ (c_0, c_1, c_2, \ldots, c_{n-1}) \in \mathbb{C} \quad \Rightarrow \quad (c_{n-1}, c_0, c_1, \ldots, c_{n-2}) \in \mathbb{C} \]
Zeros of a cyclic LRC code

Let $\mathcal{C}$ be a $q$-ary cyclic code

$$(c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C} \quad \rightarrow \quad c(x) = \sum_{i=0}^{n-1} c_i x^i$$

$\mathcal{C}$ is an ideal in the ring $F_q[x]/(x^n - 1)$; $\mathcal{C} = \langle g(x) \rangle$, where $g(x)$ is the generator polynomial of $\mathcal{C}$

**Definition:** Zeros of the code $\mathcal{C} = \text{zeros of } g(x)$
Zeros of a cyclic LRC code

Let $\mathcal{C}$ be a $q$-ary cyclic code

$$(c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C} \rightarrow c(x) = \sum_{i=0}^{n-1} c_i x^i$$

$\mathcal{C}$ is an ideal in the ring $F_q[x]/(x^n - 1)$; $\mathcal{C} = \langle g(x) \rangle$, where $g(x)$ is the generator polynomial of $\mathcal{C}$

**Definition:** Zeros of the code $\mathcal{C} = \text{zeros of } g(x)$

<table>
<thead>
<tr>
<th>Generator matrix</th>
<th>Parity-check matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G = \begin{pmatrix} 1 &amp; 1 &amp; 1 &amp; \ldots &amp; 1 \ 1 &amp; \alpha &amp; \alpha^2 &amp; \ldots &amp; \alpha^{n-1} \ \vdots &amp; \vdots &amp; \vdots &amp; \ddots &amp; \vdots \ 1 &amp; \alpha^{k-1} &amp; \alpha^{2(k-1)} &amp; \ldots &amp; \alpha^{(k-1)(n-1)} \end{pmatrix}$</td>
<td>$H = \begin{pmatrix} 1 &amp; \alpha &amp; \alpha^2 &amp; \ldots &amp; \alpha^{n-1} \ 1 &amp; \alpha^2 &amp; \alpha^2 \cdot 2 &amp; \ldots &amp; \alpha^2(n-1) \ \vdots &amp; \vdots &amp; \vdots &amp; \ddots &amp; \vdots \ 1 &amp; \alpha^{n-k} &amp; \alpha^{2(n-k)} &amp; \ldots &amp; \alpha^{(n-k)(n-1)} \end{pmatrix}$</td>
</tr>
</tbody>
</table>

$\mathcal{C}$ is a cyclic code with zeros $\alpha, \alpha^2, \ldots, \alpha^{n-k}$
Cyclic codes: Example

- RS code $C$ of length $n = 15, k = 8, d = 8, q = 2^4$
  
  Zeros of $C$: $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7$
  
  Generator polynomial $g(x) = \prod_{i=1}^{7} (x - \alpha^i)$, $\text{dim}(C) = n - \deg(g) = 8$

BCH bound: $d(C) \geq \text{number of consecutive 0's} + 1$
Cyclic codes: Example

- RS code $\mathcal{C}$ of length $n = 15$, $k = 8$, $d = 8$, $q = 2^4$
- Zeros of $\mathcal{C}$: $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7$
- Generator polynomial $g(x) = \prod_{i=1}^{7} (x - \alpha^i)$, $\dim(\mathcal{C}) = n - \deg(g) = 8$

BCH bound: $d(\mathcal{C}) \geq$ number of consecutive 0's + 1

- Now suppose that $\mathcal{C}$ has zeros

$$\{\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7\} \cup \{\alpha, \alpha^4, \alpha^7, \alpha^{10}, \alpha^{13}\}.$$ 

The distance is still $d = 8$, and we get locality $r = 2$
Cyclic codes: Example

- RS code $\mathcal{C}$ of length $n = 15$, $k = 8$, $d = 8$, $q = 2^4$
  
  Zeros of $\mathcal{C}$: $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7$
  
  Generator polynomial $g(x) = \prod_{i=1}^{7} (x - \alpha^i)$, $\dim(\mathcal{C}) = n - \deg(g) = 8$

BCH bound: $d(\mathcal{C}) \geq$ number of consecutive 0’s + 1

- Now suppose that $\mathcal{C}$ has zeros $\{\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7\} \cup \{\alpha, \alpha^4, \alpha^7, \alpha^{10}, \alpha^{13}\}$.

  The distance is still $d = 8$, and we get locality $r = 2$

  Indeed,

  $$f_a(x) = a_1 + a_2x + a_3x^3 + a_4x^4 + a_5x^6 + a_6x^7$$
Cyclic codes: Example

- RS code $\mathcal{C}$ of length $n = 15$, $k = 8$, $d = 8$, $q = 2^4$
  
  Zeros of $\mathcal{C}$: $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7$
  
  Generator polynomial $g(x) = \prod_{i=1}^{7} (x - \alpha^i)$, $\dim(\mathcal{C}) = n - \deg(g) = 8$

  \[\text{BCH bound: } d(\mathcal{C}) \geq \text{number of consecutive 0's} + 1\]

- Now suppose that $\mathcal{C}$ has zeros

  \[
  \{\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7\} \cup \{\alpha, \alpha^4, \alpha^7, \alpha^{10}, \alpha^{13}\}.
  \]

  The distance is still $d = 8$, and we get locality $r = 2$

  Indeed,

  \[f_a(x) = a_1 + a_2x + a_3x^3 + a_4x^4 + a_5x^6 + a_6x^7\]

  The rows of $G$ are $1, \alpha, \alpha^3, \alpha^4, \alpha^6, \alpha^7$
Cyclic codes: Example

- RS code $C$ of length $n = 15, k = 8, d = 8, q = 2^4$
  - Zeros of $C$: $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7$
  - Generator polynomial $g(x) = \prod_{i=1}^{7} (x - \alpha^i)$, $\dim(C) = n - \deg(g) = 8$

BCH bound: $d(C) \geq \text{number of consecutive 0's} + 1$

- Now suppose that $C$ has zeros

$\{\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7\} \cup \{\alpha, \alpha^4, \alpha^7, \alpha^{10}, \alpha^{13}\}$

The distance is still $d = 8$, and we get locality $r = 2$

Indeed,

$f_a(x) = a_1 + a_2x + a_3x^3 + a_4x^4 + a_5x^6 + a_6x^7$

The rows of $G$ are $1, \alpha, \alpha^3, \alpha^4, \alpha^6, \alpha^7$

$k = 6; d = 8 = n - k \frac{r+1}{r} + 2$
Cyclic LRC codes

- Consider a cyclic code of length $n | (q - 1)$ given by evaluations $\text{eval}(f_a(x))$ of the polynomials of the form $f_a(x) = \sum_{i=0}^{k \cdot (r+1) - 2} a_i x^i, a_i \in \mathbb{F}_q$

- The zeros of the code are a union of two (overlapping) subsets:
  - Subset $D$ gives the distance, $|D| = n - \frac{k}{r} (r + 1) + 1$
  - Subset $L$ supports locality, $|L \setminus D| = \frac{k}{r} - 1$

- The code is Singleton-optimal by the BCH bound
Cyclic LRC codes

- Consider a cyclic code of length \( n \mid (q - 1) \) given by evaluations \( \text{eval}(f_a(x)) \) of the polynomials of the form
  \[
  f_a(x) = \sum_{i=0 \atop i \neq r \mod (r+1)}^{k \cdot \frac{r}{r+1} - 2} a_i x^i, \quad a_i \in \mathbb{F}_q
  \]

- The zeros of the code are a union of two (overlapping) subsets:
  - subset \( D \) gives the distance, \( |D| = n - \frac{k}{r} (r + 1) + 1 \)
  - subset \( L \) supports locality, \( |L \setminus D| = \frac{k}{r} - 1 \)

- The code is Singleton-optimal by the BCH bound

\[
\text{zeros of } C \text{ are of the form } \alpha^j,
\]

\[
j \in \{1, 2, \ldots, n - \frac{k}{r} (r + 1) + 1\}; \{n - (\frac{k}{r} - 1)(r + 1) + 1, n - (\frac{k}{r} - 2)(r + 1) + 1, \ldots, n - r\} \]
Let $0 \leq l \leq r$, $\nu = n/(r + 1)$. Consider the $\nu \times n$ matrix (the zeros from $L$)

\[
\mathcal{H}' = \begin{pmatrix}
1 & \alpha^l & \alpha^{2l} & \ldots & \alpha^{(n-1)(r+1)+l} \\
1 & \alpha^{(r+1)+l} & \alpha^{2((r+1)+l)} & \ldots & \alpha^{(n-1)((r+1)+l)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \alpha^{(\nu-1)(r+1)+l} & \alpha^{2((\nu-1)(r+1)+l)} & \ldots & \alpha^{(n-1)((\nu-1)(r+1)+l)}
\end{pmatrix}
\]
Let $0 \leq l \leq r$, $\nu = n/(r + 1)$. Consider the $\nu \times n$ matrix (the zeros from $L$)

$$
\mathcal{H}' = \begin{pmatrix}
1 & \alpha^l & \alpha^{2l} & \ldots & \alpha^{(n-1)(r+1)+l} \\
1 & \alpha^{(r+1)+l} & \alpha^{2((r+1)+l)} & \ldots & \alpha^{(n-1)((r+1)+l)} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
1 & \alpha^{(\nu-1)(r+1)+l} & \alpha^{2((\nu-1)(r+1)+l)} & \ldots & \alpha^{(n-1)((\nu-1)(r+1)+l)} \\
\end{pmatrix}
$$

The row space of $\mathcal{H}'$ contains all the cyclic shifts of the $n$-dimensional vector of weight $r + 1$

$$
\nu = \begin{pmatrix}
1 & 0 & \ldots & 0 & \alpha^{l
\nu} & 0 & \ldots & 0 & \alpha^{2l\nu} & 0 & \ldots & 0 & \ldots & \alpha^{r\nu} & 0 & \ldots & 0
\end{pmatrix}, \quad \nu = n(r+1)
$$
Let $0 \leq l \leq r$, $\nu = n/(r + 1)$. Consider the $\nu \times n$ matrix (the zeros from $L$)

$$
\mathcal{H}' = \begin{pmatrix}
1 & \alpha^l & \alpha^{2l} & \ldots & \alpha^{(n-1)(r+1)+l} \\
1 & \alpha^{(r+1)+l} & \alpha^{2((r+1)+l)} & \ldots & \alpha^{(n-1)((r+1)+l)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \alpha^{(\nu-1)(r+1)+l} & \alpha^{2((\nu-1)(r+1)+l)} & \ldots & \alpha^{(n-1)((\nu-1)(r+1)+l)}
\end{pmatrix}
$$

The row space of $\mathcal{H}'$ contains all the cyclic shifts of the $n$-dimensional vector of weight $r + 1$

$$
\nu = (1 \underbrace{0 \ldots 0}_{\nu-1} \alpha^l \underbrace{0 \ldots 0}_{\nu-1} \alpha^{2l} \underbrace{0 \ldots 0}_{\nu-1} \ldots \underbrace{0 \ldots 0}_{\nu-1} \alpha^{r\nu} \underbrace{0 \ldots 0}_{\nu-1}), \nu = n(r + 1)
$$

The code $C$ has the parity-check matrix $H = \mathcal{H} \cup \mathcal{H}'$, where

- $\mathcal{H}$ is formed by the rows in $D$
- $\mathcal{H}'$ is formed by the rows in $L \setminus D$
Main ideas II

To obtain a code over a small field (e.g., $\mathbb{F}_p$) take a subfield subcode of the code $C$, i.e.,

$$\mathcal{D} = C \cap (\mathbb{F}_p)^n$$
Main ideas II

To obtain a code over a small field (e.g., $\mathbb{F}_p$) take a subfield subcode of the code $\mathcal{C}$, i.e.,

$$D = \mathcal{C} \cap (\mathbb{F}_p)^n$$

Analysis is based on Delsarte’s theorem: $\mathcal{C} \in \mathbb{F}_q^n$, $q = p^m$

$$\mathcal{C} \quad \longleftrightarrow \quad \mathcal{C}^\perp$$

$$\bigcap (\mathbb{F}_p)^n \quad \quad \quad T_m$$

$$D \quad \longleftrightarrow \quad D^\perp = T_m(\mathcal{C}^\perp)$$

(Delsarte ’75; Sidelnikov ’72)

Trace $T_m : \mathbb{F}_{p^m} \rightarrow \mathbb{F}_p$: $T_m(a) = a + a^p + \cdots + a^{p^{m-1}}$

- Since $D$ is cyclic, $r = d^\perp - 1$
- We would like small $r$, i.e., upper estimates of $d^\perp$
- This is in contrast to classical coding where one seeks large $d^\perp$
Main ideas II

To obtain a code over a small field (e.g., $\mathbb{F}_p$) take a subfield subcode of the code $C$, i.e.,

$$D = C \cap (\mathbb{F}_p)^n$$

Analysis is based on Delsarte’s theorem: $C \in \mathbb{F}_q^n, q = p^m$

$$C \iff C^\perp$$

$$\bigcap (\mathbb{F}_p)^n \iff T_m$$

$$D \iff D^\perp = T_m(C^\perp)$$

(Delsarte ’75; Sidelnikov ’72)

Trace $T_m : \mathbb{F}_{p^m} \rightarrow \mathbb{F}_p$: $T_m(a) = a + a^p + \cdots + a^{p^{m-1}}$

Further ideas involve

- Theory of irreducible (minimal) cyclic codes
- Results on upper bounds on the distance of a cyclic code in terms of its zeros
A sampling of results

In a number of cases it is possible to estimate the locality and the number of recovery sets of codes.

Let $\alpha$ be an $n$th root of unity, let $(2^t - 1)|n$ for some $t$.

Let $D$ be an $[n, k]$ binary linear cyclic code whose defining set of zeros contains the group $\langle \alpha^{2^t - 1} \rangle$. Then the locality of $D$ satisfies

$$r \leq 2^{t-1} - 1,$$

and each symbol has $\geq 2^{t-1}$ recovery sets.

For instance, we have the following binary LRC codes:

<table>
<thead>
<tr>
<th>$n$</th>
<th>$k$</th>
<th>$d$</th>
<th>$Z(D)$</th>
<th>$t$</th>
<th>$r$</th>
<th>$Z(D^\perp)$</th>
<th>$d^\perp$</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>20</td>
<td>3</td>
<td>{1, 15}</td>
<td>3</td>
<td>$r \leq 3$</td>
<td>{0, 1, 7, 15}</td>
<td>4</td>
</tr>
<tr>
<td>45</td>
<td>33</td>
<td>3</td>
<td>{1}</td>
<td>4</td>
<td>$r \leq 7$</td>
<td>{0, 1, 3, 5, 9, 15, 21}</td>
<td>8</td>
</tr>
<tr>
<td>27</td>
<td>7</td>
<td>6</td>
<td>{1, 9}</td>
<td>2</td>
<td>$r = 1$</td>
<td>{0, 3}</td>
<td>2</td>
</tr>
<tr>
<td>63</td>
<td>36</td>
<td>3</td>
<td>{1, 9, 11, 15, 23}</td>
<td>3</td>
<td>$r \leq 3$</td>
<td>{0, 1, 7, 9, 11, 15, 21, 23}</td>
<td>4</td>
</tr>
</tbody>
</table>
Some open questions

- In several examples the bounds on locality (dual distance) give tight results. Is it possible to characterize the cases in which the bounds are tight?
- Find the number of recovery sets per symbol for families of cyclic codes.
In this part we discuss another approach to constructing long codes over small alphabets.

RS codes can be viewed as codes on the (affine) line, and can be extended to codes on algebraic curves. The constructions becomes more technical, and we proceed by example.
Consider the set of pairs \((x, y) \in \mathbb{F}_9\) that satisfy the equation \(x^3 + x = y^4\)
LRC codes on curves

Consider the set of pairs \((x, y) \in \mathbb{F}_9\) that satisfy the equation \(x^3 + x = y^4\)

Affine points of the Hermitian curve \(X\) over \(\mathbb{F}_9\); \(\alpha^2 = \alpha + 1\)
Hermitian codes

\[ g : \mathcal{X} \rightarrow \mathbb{P}^1 \]

\[ (x, y) \mapsto y \]

Space of functions \( V := \langle 1, y, y^2, x, xy, xy^2 \rangle \)

\( A = \{ \text{Affine points of the Hermitian curve over } \mathbb{F}_9 \}; n = 27, k = 6 \)

\( C : V \rightarrow \mathbb{F}_9^n \)
Hermitian codes

\[ g : \mathcal{X} \rightarrow \mathbb{P}^1 \]

\[ (x, y) \mapsto y \]

Space of functions \( V := \langle 1, y, y^2, x, xy, xy^2 \rangle \)

\( A = \{ \text{Affine points of the Hermitian curve over } \mathbb{F}_9 \}; \ n = 27, k = 6 \)

\[ C : V \rightarrow \mathbb{F}_9^n \]

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

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

\[ F(0, 0) = 1 \text{ etc.} \]
LRC codes on curves

\[
\begin{array}{cccccc}
\alpha^7 & \alpha & \alpha^7 & \alpha^5 & 0 \\
\alpha^6 & \alpha^2 \\
\alpha^5 & \alpha^6 & \alpha^4 & \alpha^2 & 0 \\
\alpha^4 & \alpha^7 & \alpha^3 & \alpha^5 & \alpha^5 \\
x & \alpha^3 & \alpha^3 & \alpha^7 & \alpha & \alpha \\
\alpha^2 & \alpha^3 \\
\alpha & 0 & 0 & 0 & 0 \\
1 & 1 & \alpha^6 & \alpha^4 & 0 \\
0 & 1 \\
0 & 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 \\
y
\end{array}
\]
Let $P = (\alpha, 1)$ be the erased location.
Local recovery with Hermitian codes

<table>
<thead>
<tr>
<th></th>
<th>(\alpha^7)</th>
<th>(\alpha)</th>
<th>(\alpha^7)</th>
<th>(\alpha^5)</th>
<th>(0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha^6)</td>
<td>(\alpha^2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\alpha^5)</td>
<td></td>
<td>(\alpha^6)</td>
<td>(\alpha^4)</td>
<td>(\alpha^2)</td>
<td>(0)</td>
</tr>
<tr>
<td>(\alpha^4)</td>
<td></td>
<td></td>
<td>(\alpha^7)</td>
<td>(\alpha^3)</td>
<td>(\alpha^5)</td>
</tr>
<tr>
<td>(x) (\alpha^3)</td>
<td>(\alpha^3)</td>
<td>(\alpha^7)</td>
<td>(\alpha)</td>
<td>(\alpha)</td>
<td></td>
</tr>
<tr>
<td>(\alpha^2)</td>
<td>(\alpha^3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\alpha)</td>
<td>?</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>(\alpha^6)</td>
<td>(\alpha^4)</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>(\alpha)</td>
<td>(\alpha^2)</td>
<td>(\alpha^3)</td>
<td>(\alpha^4)</td>
</tr>
</tbody>
</table>

Let \(P = (\alpha, 1)\) be the erased location. Recovery set \(I_P = \{(\alpha^4, 1), (\alpha^3, 1)\}\)
Find \(f(x) : f(\alpha^4) = \alpha^7, f(\alpha^3) = \alpha^3\)

\[\Rightarrow f(x) = \alpha x - \alpha^2\]
Let $P = (\alpha, 1)$ be the erased location. Recovery set $I_P = \{(\alpha^4, 1), (\alpha^3, 1)\}$

Find $f(x) : f(\alpha^4) = \alpha^7, f(\alpha^3) = \alpha^3$

$$\Rightarrow f(x) = \alpha x - \alpha^2$$
$$f(\alpha) = 0 = F(P)$$
Let $P = (\alpha, 1)$ be the erased location. Recovery set $I_P = \{ (\alpha^4, 1), (\alpha^3, 1) \}$

Find $f(x) : f(\alpha^4) = \alpha^7$, $f(\alpha^3) = \alpha^3$

\[ \Rightarrow f(x) = \alpha x - \alpha^2 \]

\[ f(\alpha) = 0 = F(P) \]

Computations can be done using GAP or Magma
Hermitian codes

\[ q = q_0^2, \quad q_0 \text{ prime power} \]
Hermitian codes

\[ q = q_0^2, \ q_0 \text{ prime power} \]

\[ X: x^{q_0} + x = y^{q_0+1} \]
$q = q_0^2$, $q_0$ prime power

$\mathcal{X} : x^{q_0} + x = y^{q_0+1}$

$\mathcal{X}$ has $q_0^3 = q^{3/2}$ points in $\mathbb{F}_q$
Hermitian codes

$q = q_0^2$, $q_0$ prime power

$$\mathcal{X} : x^{q_0} + x = y^{q_0 + 1}$$

$\mathcal{X}$ has $q_0^3 = q^{3/2}$ points in $\mathbb{F}_q$

We obtain a family of $q$-ary codes of length $n = q_0^3$,

$$k = (t + 1)(q_0 - 1), \quad d \geq n - tq_0 - (q_0 - 2)(q_0 + 1)$$

with locality $r = q_0 - 1$. 
Geometric view of the construction

We take $g : \mathcal{X} \rightarrow y = \mathbb{P}^1$, $g(P) = g(x, y) := y$

| $\alpha^7$ |   |   |   |   |   |
| $\alpha^6$ |   |   |   |   |   |
| $\alpha^5$ |   |   |   |   |   |
| $\alpha^4$ |   |   |   |   |   |
| $\alpha^3$ |   |   |   |   |   |
| $\alpha^2$ |   |   |   |   |   |
| $\alpha$   |   |   |   |   |   |
| 1           |   |   |   |   |   |
| 0           |   |   |   |   |   |

Projection on $y$
Geometric view of the construction

We take $g : \mathcal{X} \rightarrow \mathbb{Y} = \mathbb{P}^1$, $g(P) = g(x, y) := y$

Projection on $y$

Space of functions $V := \langle 1, y, y^2, x, xy, xy^2 \rangle$

Since $y$ is constant on the fibers (recovery sets), we get back to the univariate polynomial interpolation
We take $g : \mathcal{X} \rightarrow \mathbb{Y} = \mathbb{P}^1$, $g(P) = g(x, y) := y$

Space of functions $V := \langle 1, y, y^2, x, xy, xy^2 \rangle$

Since $y$ is constant on the fibers (recovery sets), we get back to the univariate polynomial interpolation

It is also possible to take $g(P) = x$ (projection on $x$); we obtain LRC codes with locality $q_0$
In the RS-like construction, $\mathcal{X} = y = \mathbb{P}^1$
General construction: Technical details

**Map of curves**

$X, Y$ smooth projective absolutely irreducible curves over $\mathbb{k}$

$g : X \to Y$

rational separable map of degree $r + 1$
Map of curves

$X, Y$ smooth projective absolutely irreducible curves over $\mathbb{L}_k$

$g : X \to Y$

rational separable map of degree $r + 1$

Lift the points of $Y$

$S = \{P_1, \ldots, P_s\} \subseteq Y(\mathbb{L}_k)$. Partition of points:

$A := g^{-1}(S) = \{P_{ij}, i = 0, \ldots, r, j = 1, \ldots, s\} \subseteq X(\mathbb{L}_k)$

such that $g(P_{ij}) = P_j$ for all $i, j$
General construction: Technical details

Map of curves

$X$, $Y$ smooth projective absolutely irreducible curves over $\mathbb{F}_q$

$g : X \rightarrow Y$

rational separable map of degree $r + 1$

Lift the points of $Y$

$S = \{P_1, \ldots, P_s\} \subset Y(\mathbb{F}_q)$. Partition of points:

$A := g^{-1}(S) = \{P_{ij}, i = 0, \ldots, r, j = 1, \ldots, s\} \subset X(\mathbb{F}_q)$

such that $g(P_{ij}) = P_j$ for all $i, j$

Let $x \in \mathbb{F}_q(X)$ be such that $\mathbb{F}_q(X) = \mathbb{F}_q(Y)(x)$, and let $\deg x = h$ as a projection $x : X \rightarrow \mathbb{P}^1_{\mathbb{F}_q}$
Let $Q_\infty \subset \pi^{-1}(\infty)$, $\deg Q_\infty = \ell \geq 1$

Let $\mathcal{L}(Q_\infty) = \langle f_1, \ldots, f_m \rangle$, $m \geq \ell - g_Y + 1$

**Function space**

$$V := \langle f_j x^i, i = 0, \ldots, r - 1; j = 1, \ldots, m \rangle$$
General construction, II

Let $Q_\infty \subset \pi^{-1}(\infty), \deg Q_\infty = \ell \geq 1$

Let $\mathcal{L}(Q_\infty) = \langle f_1, \ldots, f_m \rangle, m \geq \ell - g_Y + 1$

Function space

$$V := \langle f_j x^i, i = 0, \ldots, r - 1; j = 1, \ldots, m \rangle$$

The code $\mathcal{C}$ is an image of the map

$$e := ev_A : V \rightarrow \mathbb{K}^{(r+1)s}$$

$$F \mapsto (F(P_{ij}), i = 0, \ldots, r, j = 1, \ldots, s)$$
General construction, II

Let $Q_\infty \subset \pi^{-1}(\infty)$, $\deg Q_\infty = \ell \geq 1$

Let $\mathcal{L}(Q_\infty) = \langle f_1, \ldots, f_m \rangle$, $m \geq \ell - g_Y + 1$

Function space

$$V := \left\langle f_j x^i, i = 0, \ldots, r - 1; j = 1, \ldots, m \right\rangle$$

The code $C$ is an image of the map

$$e := ev_A : V \longrightarrow \mathbb{K}^{(r+1)s}$$

$$F \mapsto (F(P_{ij}), i = 0, \ldots, r, j = 1, \ldots, s)$$

**Theorem:** The subspace $C(D, g) \subset \mathbb{F}_q$ forms an $(n, k, r)$ linear LRC code with the parameters

$$\begin{align*}
n &= (r + 1)s \\
k &= rm \geq r(\ell - g_Y + 1) \\
d &\geq n - \ell(r + 1) - (r - 1)h
\end{align*}$$

provided that the right-hand side of the inequality for $d$ is a positive integer.
Asymptotically good sequences of codes

In classic coding problem, codes on algebraic curves give rise to some excellent code families, in particular, improving the asymptotic Gilbert-Varshamov bound on the parameters

(Tsfasman-Vlăduț-Zink '81)
Asymptotically good sequences of codes

Let \( q = q_0^2 \), where \( q_0 \) is a prime power. Take Garcia-Stichtenoth towers of curves:

\[
x_0 := 1; \quad X_1 := \mathbb{P}^1, \quad \mathbb{L}(X_1) = \mathbb{L}(x_1); \\
X_l : x_l^{q_0} + z_l = x_{l-1}^{q_0+1}, \quad x_{l-1} := \frac{z_{l-1}}{x_{l-2}} \in \mathbb{L}(X_{l-1}) \quad \text{(if } l \geq 3)\]

There exist families of \( q \)-ary LRC codes with locality \( r \) whose rate and relative distance satisfy

\[
R \geq \frac{r}{r+1} \left( 1 - \delta - \frac{3}{\sqrt{q} + 1} \right), \quad r = \sqrt{q} - 1
\]

\[
R \geq \frac{r}{r+1} \left( 1 - \delta - \frac{2\sqrt{q}}{q - 1} \right), \quad r = \sqrt{q}
\]
Asymptotically good sequences of codes

Let $q = q_0^2$, where $q_0$ is a prime power. Take Garcia-Stichtenoth towers of curves:

$$x_0 := 1; \quad X_1 := \mathbb{P}^1, \mathbb{L}(X_1) = \mathbb{L}(x_1);$$

$$X_l : z_l^{q_0} + z_l = x_{l-1}^{q_0+1}, \quad x_{l-1} := \frac{z_{l-1}}{x_{l-2}} \in \mathbb{L}(X_{l-1}) \text{ (if } l \geq 3)$$

There exist families of $q$-ary LRC codes with locality $r$ whose *rate and relative distance* satisfy

$$R \geq \frac{r}{r+1} \left(1 - \delta - \frac{3}{\sqrt{q} + 1}\right), \quad r = \sqrt{q} - 1$$

$$R \geq \frac{r}{r+1} \left(1 - \delta - \frac{2\sqrt{q}}{q-1}\right), \quad r = \sqrt{q}$$

*) Recall the TVZ '81 bound without locality: $R \geq 1 - \delta - \frac{1}{\sqrt{q}-1}$
LRC codes on curves better than the GV bound

The asymptotic GV bound can be improved for any given (constant) $r$ for all $q$ greater than some value.

Constructions of LRC codes
LRC codes on curves better than the GV bound

The asymptotic GV bound can be improved for any given (constant) \( r \) for all \( q \) greater than some value.

Constructions of LRC codes
Extensions and open questions

Extensions

- LRC codes that correct locally multiple erasures
- LRC codes on curves with availability
- Adjusting the locality value $r$

Open questions

- Constructions of LRC codes from known families of good curves
- Parameters of subfield subcodes of AG LRC codes
- Automorphism groups of curves and codes with availability
**References**


Erasure (and LRC) codes in practice

A brief overview
Before LRC codes

**Replication:** Google (x3, x27), FB (x3), HDFS, others
Before LRC codes

**Replication:** Google (x3, x27), FB (x3), HDFS, others

**RS codes, e.g., [6,4] RS codes:** RAID; used by FB, others
Replication: Google (x3, x27), FB (x3), HDFS, others

RS codes, e.g., [6,4] RS codes: RAID; used by FB, others

Array-type codes: RAID 6 (2 parities per block); IBM (EVENODD, X-Code)
Before LRC codes

Replication: Google (x3, x27), FB (x3), HDFS, others

RS codes, e.g., [6,4] RS codes: RAID; used by FB, others

Array-type codes: RAID 6 (2 parities per block); IBM (EVENODD, X-Code)

“There are two kinds of erasure codes implemented in Raid: XOR code and Reed-Solomon code. [...] The replication on the source file can be reduced to 1 when using Reed-Solomon without losing data safety. The downside of having only one replica of a block is that reads of a block have to go to a single machine, reducing parallelism. Thus Reed-Solomon should be used on data that is not supposed to be used frequently.”

http://wiki.apache.org/hadoop/HDFS-RAID

LRC codes in practice
Applications of erasure codes

Systems using existing solutions:
- Amazon Web Services (AWS) + Glacier archiving service (previously S3); on the market
- Dropbox (moved its storage from AWS to own system)
- Google Colossus distributed FS ([9,6] RS codes; OSDI 2010)
- Facebook F4 ([14,10] RS code; OSDI 2014)
- Ceph storage platform (ceph.com)

Innovative proposals, erasure codes with some form of locality:
- HDFS-RAID, including HDFS Xorbas (FB)
- Microsoft Azure storage (VLDB Endowment, 2013)
- HACFS, adaptive coding for HDFS (IBM) (FAST ’15)
- Glacier (NSDI ’05), Petal (ASPLOS ’96), Weaver (FAST ’05), Stair codes (FAST ’14), CORE (MSST 2013), ...

Proposals based on regenerating codes (Many more)
Applications of erasure codes

Systems using existing solutions:

Amazon Web Services (AWS) + Glacier archiving service (previously S3); on the market
Dropbox (moved its storage from AWS to own system)
Google Colossus distributed FS ([9,6] RS codes; OSDI 2010)
Facebook F4 ([14,10] RS code; OSDI 2014)
Ceph storage platform (ceph.com)

Innovative proposals, erasure codes with some form of locality:
HDFS-RAID, including HDFS Xorbas (FB)
Microsoft Azure storage (VLDB Endowment, 2013)
HACFS, adaptive coding for HDFS (IBM) (FAST ’15)
Glacier (NSDI ’05), Petal (ASPLOS ’96), Weaver (FAST ’05), Stair codes (FAST ’14), CORE (MSST 2013), ...

Proposals based on regenerating codes (Many more)
Applications of erasure codes

Systems using existing solutions:

- Amazon Web Services (AWS) + Glacier archiving service (previously S3); on the market
Applications of erasure codes

Systems using existing solutions:

- Amazon Web Services (AWS) + Glacier archiving service (previously S3); on the market
- Dropbox (moved its storage from AWS to own system)
Applications of erasure codes

Systems using existing solutions:

- Amazon Web Services (AWS) + Glacier archiving service (previously S3); on the market
- Dropbox (moved its storage from AWS to own system)
- Google Colossus distributed FS ([9,6] RS codes; OSDI2010)
Applications of erasure codes

Systems using existing solutions:

- Amazon Web Services (AWS) + Glacier archiving service (previously S3); on the market
- Dropbox (moved its storage from AWS to own system)
- Google Colossus distributed FS ([9,6] RS codes; OSDI 2010)
- Facebook F4 ([14,10] RS code; OSDI 2014)
Applications of erasure codes

Systems using existing solutions:

- Amazon Web Services (AWS) + Glacier archiving service (previously S3); on the market
- Dropbox (moved its storage from AWS to own system)
- Google Colossus distributed FS ([9,6] RS codes; OSDI 2010)
- Facebook F4 ([14,10] RS code; OSDI 2014)
- Ceph storage platform (ceph.com)

Innovative proposals, erasure codes with some form of locality:

- HDFS-RAID, including HDFS Xorbas (FB)
- Microsoft Azure storage (VLDB Endowment, 2013)
- HACFS, adaptive coding for HDFS (IBM) (FAST '15)
- Glacier (NSDI '05), Petal (ASPLOS '96), Weaver (FAST '05), Stair codes (FAST '14), CORE (MSST 2013), ...
- Proposals based on regenerating codes (Many more)
Applications of erasure codes

Systems using existing solutions:

- Amazon Web Services (AWS) + Glacier archiving service (previously S3); on the market
- Dropbox (moved its storage from AWS to own system)
- Google Colossus distributed FS ([9,6] RS codes; OSDI 2010)
- Facebook F4 ([14,10] RS code; OSDI 2014)
- Ceph storage platform (ceph.com)

Innovative proposals, erasure codes with some form of locality:
Applications of erasure codes

Systems using existing solutions:
- Amazon Web Services (AWS) + Glacier archiving service (previously S3); on the market
- Dropbox (moved its storage from AWS to own system)
- Google Colossus distributed FS ([9,6] RS codes; OSDI 2010)
- Facebook F4 ([14,10] RS code; OSDI 2014)
- Ceph storage platform (ceph.com)

Innovative proposals, erasure codes with some form of locality:
- HDFS-RAID, including HDFS Xorbas (FB)
Applications of erasure codes

Systems using existing solutions:

- Amazon Web Services (AWS) + Glacier archiving service (previously S3); on the market
- Dropbox (moved its storage from AWS to own system)
- Google Colossus distributed FS ([9,6] RS codes; OSDI2010)
- Facebook F4 ([14,10] RS code; OSDI 2014)
- Ceph storage platform (ceph.com)

Innovative proposals, erasure codes with some form of locality:

- HDFS-RAID, including HDFS Xorbas (FB)
- Microsoft Azure storage (VLDB Endowment, 2013)
Applications of erasure codes

Systems using existing solutions:
- Amazon Web Services (AWS) + Glacier archiving service (previously S3); on the market
- Dropbox (moved its storage from AWS to own system)
- Google Colossus distributed FS ([9,6] RS codes; OSDI2010)
- Facebook F4 ([14,10] RS code; OSDI 2014)
- Ceph storage platform (ceph.com)

Innovative proposals, erasure codes with some form of locality:
- HDFS-RAID, including HDFS Xorbas (FB)
- Microsoft Azure storage (VLDB Endowment, 2013)
- HACFS, adaptive coding for HDFS (IBM) (FAST ’15)
Applications of erasure codes

Systems using existing solutions:

- Amazon Web Services (AWS) + Glacier archiving service (previously S3); on the market
- Dropbox (moved its storage from AWS to own system)
- Google Colossus distributed FS ([9,6] RS codes; OSDI 2010)
- Facebook F4 ([14,10] RS code; OSDI 2014)
- Ceph storage platform (ceph.com)

Innovative proposals, erasure codes with some form of locality:

- HDFS-RAID, including HDFS Xorbas (FB)
- Microsoft Azure storage (VLDB Endowment, 2013)
- HACFS, adaptive coding for HDFS (IBM) (FAST ’15)
- Glacier (NSDI ’05), Petal (ASPLOS ’96), Weaver (FAST ’05), Stair codes (FAST ’14), CORE (MSST 2013), ...
Applications of erasure codes

Systems using existing solutions:

- Amazon Web Services (AWS) + Glacier archiving service (previously S3); on the market
- Dropbox (moved its storage from AWS to own system)
- Google Colossus distributed FS ([9,6] RS codes; OSDI2010)
- Facebook F4 ([14,10] RS code; OSDI 2014)
- Ceph storage platform (ceph.com)

Innovative proposals, erasure codes with some form of locality:

- HDFS-RAID, including HDFS Xorbas (FB)
- Microsoft Azure storage (VLDB Endowment, 2013)
- HACFS, adaptive coding for HDFS (IBM) (FAST ’15)
- Glacier (NSDI ’05), Petal (ASPLOS ’96), Weaver (FAST ’05), Stair codes (FAST ’14), CORE (MSST 2013), ...
- Proposals based on regenerating codes
Applications of erasure codes

Systems using existing solutions:

- Amazon Web Services (AWS) + Glacier archiving service (previously S3); on the market
- Dropbox (moved its storage from AWS to own system)
- Google Colossus distributed FS ([9,6] RS codes; OSDI 2010)
- Facebook F4 ([14,10] RS code; OSDI 2014)
- Ceph storage platform (ceph.com)

Innovative proposals, erasure codes with some form of locality:

- HDFS-RAID, including HDFS Xorbas (FB)
- Microsoft Azure storage (VLDB Endowment, 2013)
- HACFS, adaptive coding for HDFS (IBM) (FAST ’15)
- Glacier (NSDI ’05), Petal (ASPLOS ’96), Weaver (FAST ’05), Stair codes (FAST ’14), CORE (MSST 2013), ...
- Proposals based on regenerating codes

(Many more)
Optimization of coding for storage

[Xia et al., HACFS, USENIX FAST’15]
Erasure coding in Ceph

Ceph: Object storage based free software storage platform for storing on a single cluster

Erasure coding:
uses a [5,3] MDS code

http://docs.ceph.com
The FB system stores files as redundant pieces of data distributed across different drives and data centers. The system distinguishes between cold, warm, and hot data. Hot data (Haystack): The data is replicated 3 times, two copies in one center on different racks, and the third copy in a different center.
Facebook’s storage system

Warm data are written once, accessed many times, and can be deleted but not modified (e.g., 400B photos).
Facebook’s storage system

Warm data are written once, accessed many times, and can be deleted but not modified (e.g., 400B photos).

Facebook’s F4 “Warm” BLOB storage system relies on [14, 10] RS codes. The stripe of 14 blocks is placed on 14 different nodes, different racks.

(S. Muralidhar e.a., OSDI ’14)
Facebook’s storage system

Warm data are written once, accessed many times, and can be deleted but not modified (e.g., 400B photos).

Facebook’s F4 “Warm” BLOB storage system relies on $[14, 10]$ RS codes. The stripe of 14 blocks is placed on 14 different nodes, different racks.

(S. Muralidhar e.a., OSDI ’14)

**Main issues:** Reliability and load analysis based on the models of failures in the system, and storage/system architecture

Optimizing storage, input/output, and bandwidth (the amount of data exchanges in the system).
Data placement

Replication = 3, Tolerates any 2 errors

1  2  3  5  6  7  8  9  10
1  2  3  5  6  7  8  9  10
1  2  3  5  6  7  8  9  10

Dependent Blocks

Replication = 1, Parity Length = 4, Tolerates any 4 errors

1  2  x  4  5  x  7  x  P1  P2  P3

http://www.slideshare.net/ydn/hdfs-raid-facebook
LRC codes for Hadoop file system
(M. Sathiamoorthy et al., 2013 VLDB Endowment)

[14,10] RS code with two additional *local parities*:

\[ X_1, X_2, X_3, X_4, X_5 \quad S_1 \quad X_6, X_7, X_8, X_9, X_{10} \quad S_2 \quad P_1, P_2, P_3, P_4 \]

Data \quad Data \quad RS parities

where \( S_1 = \sum_{i=1}^{5} a_i X_i \), \( S_2 = \sum_{i=1}^{5} b_i X_{i+5} \) are the local parities
HDFS RAID + LRC (HDFS-XORbas)

LRC codes for Hadoop file system
(M. Sathiamoorthy et al., 2013 VLDB Endowment)

[14,10] RS code with two additional *local parities*:

\[ S_1 = \sum_{i=1}^{5} a_i X_i, \quad S_2 = \sum_{i=1}^{5} b_i X_{i+5} \]

where \( S_1 \) and \( S_2 \) are the local parities.

Reliability analysis in terms of *mean time to data loss* using a Markov failure model.
Choose a basis of $C$ so that every 4 erasures other than $tX_1; X_2; X_3; S_1 u; tX_4; X_5; X_6; S_2 u$ are correctable (Maximally Recoverable Codes, refer to Part 5 "MR codes").

$S_1 = \alpha_{1,1}X_1 + \alpha_{1,2}X_2 + \alpha_{1,3}X_3, \quad S_2 = \alpha_{2,1}X_4 + \alpha_{2,2}X_5 + \alpha_{2,3}X_6$
Choose a basis of $C$ so that every 4 erasures other than

$$\{X_1, X_2, X_3, S_1\}, \quad \{X_4, X_5, X_6, S_2\}$$

are correctable (Maximally Recoverable Codes, refer to Part 5 “MR codes”).

- C. Huang et al., USENIX 2012: (6,2,2) code, (12,2,2) code (1.33x overhead)
- Installed in Windows 8.1 and Windows Server 2012
Adaptive coding for HDFS

Implements 2 code families:

- 6 x 5 Product codes
- LRC (12,2,2) codes

  12 data blocks; 2 local parities, 2 global parities

Optimizing overhead or recovery time

[Xia et al., HACFS, USENIX FAST’15]
Maximally recoverable codes for SSD arrays

RAID 5 or RAID 6 type architecture ($r = 1$ to 3 parities per row)

global parities
Maximally recoverable codes for SSD arrays

RAID 5 or RAID 6 type architecture ($r = 1$ or 2 parities per row)

Mixed failure model: device failure plus several sector failures
Maximally recoverable codes for SSD arrays

RAID 5 or RAID 6 type architecture ($r = 1$ or 2 parities per row)

Mixed failure model: device failure plus several sector failures

Algebraic construction of MR codes similar to rank error codes (Blaum et al., 2013)
A selection of references

- C. Huang et al., Erasure coding in Windows Azure storage, Usenix ATC, 2012
- M. Blaum et al., Partial MDS codes and applications in RAID, IT Trans. 2013
- * f4: Facebook’s Warm BLOB Storage system, 11th USENIX OSDI, 2014, 383–398
- * M. Xia et al., A tale of two erasure codes in HDFS, 13th USENIX FAST, 2015, 213–226
- A. Dimakis, LRC codes and index coding, Talk at DIMACS workshop, Dec. 2015
- J. Plank and C. Huang, Tutorial on erasure codes, USENIX FAST 2013
- XORing elephants: Erasure codes for Big Data, VLDB Endowment 2013, vol.6, 325–336
- K.V. Rashmi et al., Erasure-coded distributed storage systems (FB study), USENIX 2013
- * K.V. Rashmi et al., Jointly optimal erasure codes for I/O, storage & bandwidth, FAST 2015

* Includes overview of other industry-related solutions or proposals
Bounds on LRC codes
Problem

\( C \subset \mathbb{F}_q^n \) – LRC code of length \( n \), cardinality \( q^k \), distance \( d \), locality \( r \)

**Def:** For any \( i \in [n] \) there exists a recovery (helper) set \( R_i \subset [n]\setminus\{i\} \), \( |R_i| \leq r \) and a function \( \phi_i \) such that for any \( x \in C \)

\[ x_i = \phi_i(x_j, j \in R_i) \]

(We do not assume that \( R_i \cap R_j = \emptyset, i \neq j \))
Problem

\( \mathcal{C} \subseteq \mathbb{F}_q^n \) – LRC code of length \( n \), cardinality \( q^k \), distance \( d \), locality \( r \)

**Def:** For any \( i \in [n] \) there exists a recovery (helper) set \( \mathcal{R}_i \subseteq [n] \setminus \{i\} \), \( |\mathcal{R}_i| \leq r \) and a function \( \phi_i \) such that for any \( x \in \mathcal{C} \)

\[
x_i = \phi_i(x_j, j \in \mathcal{R}_i)
\]

(We do not assume that \( \mathcal{R}_i \cap \mathcal{R}_j = \emptyset, i \neq j \))

Given \( k \) define \( d(n, k, r, q) \) to be the largest possible distance of \( \mathcal{C} \)
Problem

\[ \mathcal{C} \subset \mathbb{F}_q^n \] LRC code of length \( n \), cardinality \( q^k \), distance \( d \), locality \( r \)

**Def:** For any \( i \in [n] \) there exists a recovery (helper) set \( \mathcal{R}_i \subset [n] \setminus \{i\} \), \( |\mathcal{R}_i| \leq r \) and a function \( \phi_i \) such that for any \( x \in \mathcal{C} \)

\[ x_i = \phi_i(x_j, j \in \mathcal{R}_i) \]

(We do not assume that \( \mathcal{R}_i \cap \mathcal{R}_j = \emptyset \), \( i \neq j \))

Given \( k \) define \( d(n, k, r, q) \) to be the largest possible distance of \( \mathcal{C} \)

Given \( d \) define \( k(n, d, r, q) \) to be the largest possible dimension of \( \mathcal{C} \)

(log-cardinality)
Problem

\[ C \subset \mathbb{F}_q^n \] – LRC code of length \( n \), cardinality \( q^k \), distance \( d \), locality \( r \)

**Def:** For any \( i \in [n] \) there exists a recovery (helper) set \( R_i \subset [n]\setminus\{i\}, |R_i| \leq r \) and a function \( \phi_i \) such that for any \( x \in C \)

\[ x_i = \phi_i(x_j, j \in R_i) \]

(We do not assume that \( R_i \cap R_j = \emptyset, i \neq j \))

Given \( k \) define \( d(n, k, r, q) \) to be the largest possible distance of \( C \)

Given \( d \) define \( k(n, d, r, q) \) to be the largest possible dimension of \( C \)

(log-cardinality)

**Bounds on** \( d, k \)?
First results

\[ \frac{k}{n} \leq \frac{r}{r + 1} \]

\[ d \leq n - k - \left\lfloor \frac{k}{r} \right\rfloor + 2 \]

(Gopalan e.a.. IEEE Trans. IT, 2013)
First results

\[
\frac{k}{n} \leq \frac{r}{r + 1}
\]

\[
d \leq n - k - \left\lceil \frac{k}{r} \right\rceil + 2
\]

(Gopalan e.a.. IEEE Trans. IT, 2013)

The bound on the rate was proved in the first part of the tutorial.
Let us prove the bound on the distance.
Main idea. Let $C$ be a $q$-ary code of length $n$, size $q^k$. The distance $d(C)$ satisfies

$$\forall S \subseteq [n]: |C_S| < q^k \quad d(C) \leq n - |S|$$

(A)

Construct a set $S$:

Let $m = \left\lfloor \frac{k-1}{r} \right\rfloor$. Put $T := \emptyset, L := \emptyset$. 
The Generalized Singleton bound

Main idea. Let $C$ be a $q$-ary code of length $n$, size $q^k$. The distance $d(C)$ satisfies

$$\forall S \subseteq [n]: |C_S| < q^k \quad d(C) \leq n - |S|$$

(A)

Construct a set $S$:

Let $m = \lfloor \frac{k-1}{r} \rfloor$. Put $T := \emptyset$, $L := \emptyset$.

For $j = 1, 2, \ldots, m$: Take $i_j \in (T \cup L)^c$, Put $T \leftarrow T \cup R_{i_j}$, $L \leftarrow L \cup \{i_j\}$ ($R_i$ is the recovery set for $i_j$).
The Generalized Singleton bound

Main idea. Let $C$ be a $q$-ary code of length $n$, size $q^k$. The distance $d(C)$ satisfies

$$\forall S \subseteq [n]: |C_S| < q^k \quad d(C) \leq n - |S|$$

(A)

Construct a set $S$:

Let $m = \lceil \frac{k-1}{r} \rceil$. Put $T := \emptyset$, $L := \emptyset$.

For $j = 1, 2, \ldots, m$: Take $i_j \in (T \cup L)^c$, Put $T \leftarrow T \cup R_{i_j}$, $L \leftarrow L \cup \{i_j\}$ ($R_i$ is the recovery set for $i_j$).

Clearly $|T| \leq k - 1$, so $|C_T| \leq q^{k-1}$, and $|C_{T \cup L}| \leq q^{k-1}$.
The Generalized Singleton bound

Main idea. Let \( C \) be a \( q \)-ary code of length \( n \), size \( q^k \). The distance \( d(C) \) satisfies

\[
\forall S \subseteq [n]: |C_S| < q^k \quad d(C) \leq n - |S| \quad (A)
\]

Construct a set \( S \):

Let \( m = \left\lfloor \frac{k - 1}{r} \right\rfloor \). Put \( T := \emptyset \), \( L := \emptyset \).

For \( j = 1, 2, \ldots, m \): Take \( i_j \in (T \cup L)^c \), Put \( T \leftarrow T \cup R_i, L \leftarrow L \cup \{i_j\} \) (\( R_i \) is the recovery set for \( i_j \)).

Clearly \( |T| \leq k - 1 \), so \( |C_T| \leq q^{k - 1} \), and \( |C_{T \cup L}| \leq q^{k - 1} \)

Suppose that \( |T| = k - 1 \) (if needed, add some coordinates from \([n]\setminus (T \cup L)\)) and put \( S = T \cup L \).
The Generalized Singleton bound

**Main idea.** Let $C$ be a $q$-ary code of length $n$, size $q^k$. The distance $d(C)$ satisfies

$$\forall S \subseteq [n]: |C_S| < q^k \quad d(C) \leq n - |S|$$  \hspace{1cm} (A)

**Construct a set $S$:**

Let $m = \lceil \frac{k-1}{r} \rceil$. Put $T := \emptyset, L := \emptyset$.

For $j = 1, 2, \ldots, m$: Take $i_j \in (T \cup L)^c$, Put $T \leftarrow T \cup R_{i_j}, L \leftarrow L \cup \{i_j\}$ ($R_i$ is the recovery set for $i_j$).

Clearly $|T| \leq k - 1$, so $|C_T| \leq q^{k-1}$, and $|C_{T \cup L}| \leq q^{k-1}$

Suppose that $|T| = k - 1$ (if needed, add some coordinates from $[n]\setminus(T \cup L)$ and put $S = T \cup L$.

We have $|S| = k - 1 + m = k + \lceil \frac{k}{r} \rceil - 2$ and $|C_S| < q^k$. Conclude by (A).
Shortening (alphabet-dependent) bound

Theorem (Cadambe-Mazumdar ’13)

Let \( k_q(m, d) \) be the maximum dimension of a \( q \)-ary code of length \( m \) and distance \( d \). Then

\[
k(n, d, r, q) \leq \min_{s \geq 1} \{ sr + k_q(n - s(r + 1), d) \}.
\]

Proof.

Let \( \mathcal{C} \) be a linear LRC code with distance \( d \) (the linearity assumption is easily lifted).

- Following the proof of the LRC Singleton bound, construct a subset \( J \subset [n] \) such that
  \[
  |J| = s(r + 1), \quad \dim \text{proj}_J(\mathcal{C}) \leq sr.
  \]
- Consider the shortening \( \mathcal{C}^J \) of \( \mathcal{C} \) by the coordinates in \( J \). We obtain
  \[
  \text{length } (\mathcal{C}^J) = n - s(r + 1), \quad d(\mathcal{C}^J) = d
  \]
  \[
  \dim(\mathcal{C}^J) = k - \dim \text{proj}_J(\mathcal{C}) \geq k - sr
  \]
  \[
  k_q(n - s(r + 1), d) \geq k - sr
  \]
Asymptotic problem

Bounds on codes are often considered in the asymptotics of $n \to \infty$
Let $M(n, r, \delta n)$ be the max size of a code of length $n$, distance $d$, locality $r$

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

For binary codes,

\[ R(r, \delta) \geq 1 - \min_{0 < s \leq 1} \left\{ \frac{1}{r + 1} \log_2((1 + s)^{r+1} + (1 - s)^{r+1}) - \delta \log_2 s \right\}. \]
Gilbert-Varshamov bound

For binary codes,

\[ R(r, \delta) \geq 1 - \min_{0 < s \leq 1} \left\{ \frac{1}{r + 1} \log_2 \left( (1 + s)^{r+1} + (1 - s)^{r+1} \right) - \delta \log_2 s \right\}. \]

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

\[
H = \begin{bmatrix}
11 \ldots 1 \\
11 \ldots 1 \\
\ddots \\
11 \ldots 1 \\
\end{bmatrix}
\]

(Cadambe-Mazumdar '15; Tamo-B.-Frolov '15)
Gilbert-Varshamov bound

For binary codes,

\[ R(r, \delta) \geq 1 - \min_{0 < s \leq 1} \left\{ \frac{1}{r + 1} \log_2 \left( (1 + s)^{r+1} + (1 - s)^{r+1} \right) - \delta \log_2 s \right\}. \]

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

\[
H = \begin{bmatrix}
11\ldots1 \\
11\ldots1 \\
\ddots \\
11\ldots1
\end{bmatrix}
\]

where

\[
\Pr(\{d(C) < \delta n\}) \leq \delta n q^{\frac{r}{r+1} - \frac{r}{r+1} - R} \min_{0 < s \leq 1} \frac{b(s)^{\frac{n}{r+1}}}{s^{\delta n}}
\]
Gilbert-Varshamov bound

For binary codes,

\[ R(r, \delta) \geq 1 - \min_{0 < s \leq 1} \left\{ \frac{1}{r + 1} \log_2((1 + s)^{r+1} + (1 - s)^{r+1}) - \delta \log_2 s \right\}. \]

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

\[
H = \begin{bmatrix}
11 \ldots 1 \\
\vdots \\
11 \ldots 1 \\
\end{bmatrix}
\]

\[
\Pr\{d(C) < \delta n\} \leq \delta n q^{-n\left(\frac{r}{r+1} - R\right)} \min_{0 < s \leq 1} \frac{b(s)}{s^{\delta n}}
\]

where

\[ b(s) = \frac{1}{q} ((1 + (q - 1)s)^{r+1} + (q - 1)(1 - s)^{r+1}) \]
Gilbert-Varshamov bound

For binary codes,

\[
R(r, \delta) \geq 1 - \min_{0 < s \leq 1} \left\{ \frac{1}{r + 1} \log_2 \left( (1 + s)^{r+1} + (1 - s)^{r+1} \right) - \delta \log_2 s \right\}.
\]

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

\[
H = \begin{bmatrix}
11 \ldots 1 \\
11 \ldots 1 \\
\vdots \\
11 \ldots 1 \\
\end{bmatrix}
\]

\[
Pr(\{d(C) < \delta n\}) \leq \delta q^{-n(\frac{r}{r+1} - R)} \min_{0 < s \leq 1} \frac{b(s)^{\frac{n}{r+1}}}{s^{\delta n}}
\]

\[
b(s) = \frac{1}{q} \left( (1 + (q - 1)s)^{r+1} + (q - 1)(1 - s)^{r+1} \right)
\]

(Cadambe-Mazumdar '15; Tamo-B.-Frolov '15)
Asymptotic upper bounds

Let

\[ R_q(r, \delta) = \limsup_{n \to \infty} \frac{1}{n} \log_q k(n, r, \delta n) \]

Using the Cadambe-Mazumdar bound together with classic bounds on \( k_q(m, d) \) (with no locality)

\[
R_q(r, \delta) \leq \frac{r}{r + 1} (1 - \delta), \quad 0 \leq \delta \leq 1
\]

\[
R_q(r, \delta) \leq \frac{r}{r + 1} \left( 1 - \delta \frac{q}{q - 1} \right), \quad 0 \leq \delta \leq q/(q - 1)
\]

\[
R_q(r, \delta) \leq \min_{0 \leq \tau \leq \frac{1}{r + 1}} \left\{ \tau r + (1 - \tau(r + 1))f_q\left(\frac{\delta}{1 - \tau(r + 1)}\right) \right\}
\]

where

\[
f_q(x) := h_q\left(\frac{1}{q} (q - 1 - x(q - 2) - 2\sqrt{(q - 1)x(1 - x)})\right),
\]

\[
h_q(x) := -x \log_q (x/(q - 1)) - (1 - x) \log_q (1 - x).
\]
Asymptotic bounds

Binary codes; $r = 3$
Improving GV bound using LRC codes on curves

(details in part on algebraic constructions)

(B-Tamo-Vlăduț, ’15)
Correcting more than one erasure

Our goal here is to correct locally $\geq 2$ erasures.
Correcting more than one erasure

Our goal here is to correct locally $\geq 2$ erasures.

**Def. (correcting $\rho - 1$ erasure):** $C$ is an $(n, k, r, \rho)$ LRC code if each $i \in [n]$ is contained in a subset $J_i$, $|J_i| \leq r + \rho - 1$ such that the projection of $C$ on $J_i$ forms a code with distance $\geq \rho$. 
Correcting more than one erasure

Our goal here is to correct locally $\geq 2$ erasures.

**Def.** (correcting $\rho - 1$ erasure): $C$ is an $(n, k, r, \rho)$ LRC code if each $i \in [n]$ is contained in a subset $I_i$, $|I_i| \leq r + \rho - 1$ such that the projection of $C$ on $I_i$ forms a code with distance $\geq \rho$.

Singleton-type bound:

$$d \leq n - k + 1 - \left(\left\lceil \frac{k}{r} \right\rceil - 1\right)(\rho - 1) \quad \text{(Prakash e.a., ISIT '12)}$$
Correcting more than one erasure

Our goal here is to correct locally $\geq 2$ erasures.

**Def.** (correcting $\rho - 1$ erasure): $C$ is an $(n, k, r, \rho)$ LRC code if each $i \in [n]$ is contained in a subset $J_i$, $|J_i| \leq r + \rho - 1$ such that the projection of $C$ on $J_i$ forms a code with distance $\geq \rho$.

Singleton-type bound:

$$d \leq n - k + 1 - \left(\left\lfloor \frac{k}{r} \right\rfloor - 1\right)(\rho - 1) \quad \text{(Prakash e.a., ISIT '12)}$$

GV-type bound:

- Proof by random coding, similar to the LRC GV bound
- For large $q$ the GV bound can be improved using codes on curves (refer to the part "Algebraic constructions")

Theorem (Hu, Tamo, B., ISIT’16)

Let $B(n, \rho)$ be an upper bound on the size of a code of length $n$ and distance $\rho$. If the function $B$ is log-convex in $n$, then for any $(n, k, r, \rho)$ LRC code,

$$k \leq \left\lfloor \frac{n - (d - 1)}{r + \rho - 1} \right\rfloor \log_q B(r + \rho - 1, \rho)$$

E.g., locality-dependent Plotkin bound: Let $\rho > (r + \rho - 1) \frac{q - 1}{q}$, then

$$k \leq \left\lfloor \frac{n - (d - 1)}{r + \rho - 1} \right\rfloor \log_q \frac{\rho}{\rho - \frac{q - 1}{q}(r + \rho - 1)}$$

It is also possible to derive a locality-dependent Hamming bound.
In classical coding theory, all (or most) known upper bounds on the size of codes with a given distance can be derived via linear programming (Delsarte '72). In this part we extend this approach to codes with locality.

Steps:

- Construct an association scheme for codes with locality
- Derive positivity conditions for the distance distribution (Delsarte inequalities)
- Find closed-form expressions for upper bounds
Linear programming bounds

\( C \) an \((n, k, r, \rho)\) LRC code, distance \( d \).
$C$ an $(n, k, r, \rho)$ LRC code, distance $d$.

Assume disjoint recovery groups of size $r + \rho - 1$.
Let $C$ be a $q$-ary $(n, k, r, \rho)$ LRC code with distance $d$. Define

$$T := \left\{ i = (i_1, \ldots, i_s) \mid i_1 + \ldots + i_s \geq d, \right.$$  

$$i_p \in \{0, \rho, \rho + 1, \ldots, \rho + r - 1\} \text{ for all } p = 1, \ldots, s. \right\}.$$
Let $\mathcal{C}$ be a $q$-ary $(n, k, r, \rho)$ LRC code with distance $d$. Define

$$T := \{ i = (i_1, \ldots, i_s) \mid i_1 + \ldots + i_s \geq d, \quad i_p \in \{0, \rho, \rho + 1, \ldots, \rho + r - 1\} \text{ for all } p = 1, \ldots, s\}.$$ 

Then $|\mathcal{C}| \leq 1 + \sum_{i \in T} a_i$, where $(a_i, i \in T)$ is given by the following LP problem

maximize $\sum_{i \in T} a_i$

subject to

$a_i \geq 0, \quad i \in T,$

$$\sum_{i \in T} a_i Q_{ij} \geq -K_j^{(r+\rho-1)}(0), \quad j \in [r + \rho - 1]^s \setminus \{0\}$$

(Hu et al., Proc. ISIT 2016)
Linear programming bounds

Results:

It is possible to

- Derive the Singleton, Hamming, and Plotkin bounds on \((n, k, r, \rho)\) LRC codes
- Improve the shortening bound in examples for short length (e.g., \(n = 8-20\)) by solving the LP problem
Availability (Multiple recovery sets)

**Def:** $C \subset F^n$ is an LRC code with **availability** if every coordinate $i \subset [n]$ has $t$ pairwise disjoint recovery sets $\mathcal{R}_i^{(j)}$, and $|\mathcal{R}_i^{(j)}| \leq r_j, j = 1, \ldots, t$.

For simplicity let $r_j = r, j = 1, \ldots, t$ and use the notation $(n, k, r, t)$ LRC code.
Def: $C \subseteq F^n$ is an LRC code with availability if every coordinate $i \in [n]$ has $t$ pairwise disjoint recovery sets $R_i^{(j)}$, and $|R_i^{(j)}| \leq r_j, j = 1, \ldots, t$.

For simplicity let $r_j = r, j = 1, \ldots, t$ and use the notation $(n, k, r, t)$ LRC code.

Example: (6, 3, 2, 2) LRC code with distance 3, parity-check matrix

$$H = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 0
\end{bmatrix}$$
Def: $C \subset F^n$ is an LRC code with availability if every coordinate $i \in [n]$ has $t$ pairwise disjoint recovery sets $R^{(j)}_i$, and $|R^{(j)}_i| \leq r_j, j = 1, \ldots, t$.

For simplicity let $r_j = r, j = 1, \ldots, t$ and use the notation $(n, k, r, t)$ LRC code

Example: $(6, 3, 2, 2)$ LRC code with distance 3, parity-check matrix

$$H = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 0
\end{bmatrix}$$

Constructions:

- Case $t = 2$: Two-level codes give a natural way of constructing LRC codes with $t = 2$. Examples include product codes, codes on bipartite graphs.
- Algebraic constructions of codes with $t \geq 2$ recovery sets are covered in another part of the tutorial (refer to “Introduction”)
Upper bounds on codes with availability

Extension of the Singleton bound:

\[ d \leq n - k + 2 - \left\lfloor \frac{t(k - 1) + 1}{t(r - 1) + 1} \right\rfloor \]

(Wang-Zhang '14; Rawat e.a. '14)
The parameters of a $t$-LRC code are bounded as follows:

\[
\frac{k}{n} \leq \frac{1}{\prod_{j=1}^{t}(1 + \frac{1}{jr})}
\]

\[
d \leq n - \sum_{i=0}^{t} \left\lfloor \frac{k - 1}{r^i} \right\rfloor
\]
Theorem (Tamo-B-Frolov '15)

The parameters of a $t$-LRC code are bounded as follows:

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

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

Proof idea:

- construct a directed graph $G(V, E)$, $V = [n]$, $(i, j) \in E$ iff $j$ is a part of some recovery set for $i$
- study colorings and expansion of the graph, which enables us to trace propagation of dependent vertices in $G$
Upper bounds on codes with availability

Theorem (Tamo-B-Frolov '15)

*The parameters of a* $t$-*LRC code are bounded as follows:*

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

$$d \leq n - \sum_{i=0}^{t} \left[ \frac{k - 1}{r^i} \right]$$

**Proof idea:**

- construct a directed graph $G(V, E)$, $V = [n]$, $(i, j) \in E$ iff $j$ is a part of some recovery set for $i$
- study colorings and expansion of the graph, which enables us to trace propagation of dependent vertices in $G$

- The bounds are tight in some examples, but generally are likely to be loose.
- The above bound and the bound on the previous page are incomparable (i.e., there are examples where each of them is better than the other one).
Existence of codes with availability; $t = 2$

GV-type bound for codes with two recovery sets (TBF ’16)
Existence of codes with availability; \( t = 2 \)

**GV-type bound for codes with two recovery sets (TBF ’16)**

- Let \( G \) be a regular graph of degree \( r + 1 \). Each edge is adjacent to two disjoint sets of \( r \) edges, which form 2 recovery sets
Existence of codes with availability; $t = 2$

**GV-type bound for codes with two recovery sets** (TBF ’16)

- Let $G$ be a regular graph of degree $r + 1$. Each edge is adjacent to two disjoint sets of $r$ edges, which form 2 recovery sets (bipartite graphs support recovery sets of different sizes)
GV-type bound for codes with two recovery sets (TBF ’16)

- Let $G$ be a regular graph of degree $r + 1$. Each edge is adjacent to two disjoint sets of $r$ edges, which form 2 recovery sets (bipartite graphs support recovery sets of different sizes)
- Consider the ensemble of linear codes with parity-check matrices

$$H = \begin{bmatrix}
A_{r+2} & A_{r+2} & \cdots & A_{r+2} \\
A_{r+2} & \ddots & & \\
\vdots & & \ddots & \\
A_{r+2} & & & H_L
\end{bmatrix}$$

where $A_{r+2}$ is E-V incidence matrix of the complete graph $K_{r+2}$ (or of $K_{r_1,r_2}$)

$H_L$ is a random matrix with independent entries
Existence of codes with availability; \( t = 2 \)

**GV-type bound for codes with two recovery sets** (TBF ’16)

- Let \( G \) be a regular graph of degree \( r + 1 \). Each edge is adjacent to two disjoint sets of \( r \) edges, which form 2 recovery sets (bipartite graphs support recovery sets of different sizes)

- Consider the ensemble of linear codes with parity-check matrices

\[
H = \begin{bmatrix}
A_{r+2} & A_{r+2} & \ldots & A_{r+2} \\
\vdots & & & \\
H_L
\end{bmatrix}
\]

where \( A_{r+2} \) is E-V incidence matrix of the complete graph \( K_{r+2} \) (or of \( K_{r_1,r_2} \))

- Analyze the ensemble-average distance of codes

\( H_L \) is a random matrix with independent entries.
Many recovery sets

Theorem

There exist asymptotically good sequences of $q$-ary LRC codes (fixed $q$, $n \to \infty$) with $t$ disjoint recovery sets of size $r$ for any constant $t$, $1 \leq t \leq r$.

Ideas: Existence of bipartite expanding graphs; large finite field alphabets
Many recovery sets

**Theorem**

There exist asymptotically good sequences of $q$-ary LRC codes (fixed $q$, $n \rightarrow \infty$) with $t$ disjoint recovery sets of size $r$ for any constant $t$, $1 \leq t \leq r$

**Ideas:** Existence of bipartite expanding graphs; large finite field alphabets

$t = 3; \quad r = 6$

$\delta = 0, \quad R = 1 - \frac{t}{r+1}$

Singleton bound

$$R^{(t)}(r, \delta) \leq \frac{r^t(r - 1)}{r^{t+1} - 1}(1 - \delta)$$
Open questions

- Alphabet-dependent bounds make little use of locality
- Find a good way to derive lower bounds for $t \geq 3$, upper bounds for $t \geq 2$
- Settle the corner point for $\delta = 0$ of the asymptotic bound for $t \geq 2$
- Remove the structure assumption in the LP bound
- Derive an LP (closed-form asymptotic) bound for LRC codes that correct locally one erasure
Selected references


A. Barg, I. Tamo, and S. Vlăduț, Locally recoverable codes on algebraic curves, arXiv:1603.08876; also ISIT ’15.
Maximally Recoverable Codes

Combining locality and storage efficiency
Maximally Recoverable (MR) codes

Motivation

- An $(n, k)$ code is MDS $\iff$ Any $k$ symbols of the codeword suffice for decoding
- MDS property is very important in storage
- Recovery set = a set of $r + 1$ symbols s.t. any symbol of the set can be recovered by the remaining $r$ symbols
- A set of $k$ symbols of an $(n, k, r)$ LRC code, that contains a recovery set does not suffice for decoding
- $= \Rightarrow$ LRC codes are not MDS
- Goal: LRC codes that are "close" to be MDS
An $(n, k)$ code is MDS $\iff$ Any $k$ symbols of the codeword suffice for decoding.
Maximally Recoverable (MR) codes

Motivation

- An \((n, k)\) code is MDS \iff Any \(k\) symbols of the codeword suffice for decoding

- MDS property is very important in storage
Maximally Recoverable (MR) codes

Motivation

- An \((n, k)\) code is MDS \iff Any \(k\) symbols of the codeword suffice for decoding

- MDS property is very important in storage

- Recovery set = a set of \(r + 1\) symbols s.t. any symbol of the set can be recovered by the remaining \(r\) symbols
Maximally Recoverable (MR) codes

Motivation

- An \((n, k)\) code is MDS \iff\ Any \(k\) symbols of the codeword suffice for decoding

- MDS property is very important in storage

- Recovery set = a set of \(r + 1\) symbols s.t. any symbol of the set can be recovered by the remaining \(r\) symbols

- A set of \(k\) symbols of an \((n, k, r)\) LRC code, that contains a recovery set does not suffice for decoding
Maximally Recoverable (MR) codes
Motivation

• An \( (n, k) \) code is MDS \( \iff \) Any \( k \) symbols of the codeword suffice for decoding

• MDS property is very important in storage

• Recovery set = a set of \( r + 1 \) symbols s.t. any symbol of the set can be recovered by the remaining \( r \) symbols

• A set of \( k \) symbols of an \( (n, k, r) \) LRC code, that contains a recovery set does not suffice for decoding

• \( \implies \) LRC codes are not MDS
Maximally Recoverable (MR) codes

Motivation

- An \((n, k)\) code is MDS \iff Any \(k\) symbols of the codeword suffice for decoding

- MDS property is very important in storage

- Recovery set = a set of \(r + 1\) symbols s.t. any symbol of the set can be recovered by the remaining \(r\) symbols

- A set of \(k\) symbols of an \((n, k, r)\) LRC code, that contains a recovery set does not suffice for decoding

- \(\implies\) LRC codes are not MDS

- **Goal:** LRC codes that are “close” to be MDS
$(n, k, r)$ MR codes - Definition

- An $(n, k, r)$ LRC code
- Disjoint recovery sets $R_i$, $|R_i| = r + 1$ for $i = 1, \ldots, n$
- Any set $S$ of $k$ symbols s.t. $R_i \not\subseteq S$ suffices for decoding

Remarks:
- If $\exists i, R_i \subseteq S$, $|S| = k$ then decoding of the codeword is impossible
- Terminology: MR codes = Partially MDS codes
- Equivalent definition using Matroid Theory language
An \((n, k, r)\) MR codes has the following properties:

- \((n, k, r)\) LRC code
- Disjoint recovery sets \(R_i, |R_i| = r + 1\) for \(i = 1, \ldots, n\)
- Any set \(S\) of \(k\) symbols s.t. \(R_i \notsubseteq S\) suffices for decoding

Remarks:

- If \(\exists i, R_i \subseteq S\), then decoding of the codeword is impossible
- Terminology: MR codes = Partially MDS codes
- Equivalent definition using Matroid Theory language
\((n, k, r)\) MR codes - Definition

An \((n, k, r)\) MR codes has the following properties:

- An \((n, k, r)\) LRC code
$(n, k, r)$ MR codes - Definition

An $(n, k, r)$ MR codes has the following properties:

- An $(n, k, r)$ LRC code
- Disjoint recovery sets $\mathcal{R}_i, |\mathcal{R}_i| = r + 1$ for $i = 1, ..., \frac{n}{r+1}$

Remarks:
- If $\exists i, \mathcal{R}_i \subseteq S$, then decoding of the codeword is impossible
- Terminology: MR codes = Partially MDS codes
- Equivalent definition using Matroid Theory language
(n, k, r) MR codes - Definition

An (n, k, r) MR codes has the following properties:

- An (n, k, r) LRC code
- Disjoint recovery sets \( \mathcal{R}_i, |\mathcal{R}_i| = r + 1 \) for \( i = 1, ..., \frac{n}{r+1} \)
- Any set \( S \) of \( k \) symbols s.t. \( \mathcal{R}_i \notin S \) suffices for decoding

Remarks:
- If \( \exists i, \mathcal{R}_i \subseteq S \), \( |S| = k \) then decoding of the codeword is impossible
- Terminology: MR codes = Partially MDS codes
- Equivalent definition using Matroid Theory language
(n, k, r) MR codes - Definition

An (n, k, r) MR codes has the following properties:

- An (n, k, r) LRC code
- Disjoint recovery sets $\mathcal{R}_i, |\mathcal{R}_i| = r + 1$ for $i = 1, \ldots, \frac{n}{r+1}$
- Any set $S$ of $k$ symbols s.t. $\mathcal{R}_i \not\subseteq S$ suffices for decoding

Remarks:

• If $\exists i, \mathcal{R}_i \subseteq S, |S| = k$ then decoding of the codeword is impossible
• Terminology: MR codes = Partially MDS codes
• Equivalent definition using Matroid Theory language
(n, k, r) MR codes - Definition

An (n, k, r) MR codes has the following properties:

- An (n, k, r) LRC code
- Disjoint recovery sets $\mathcal{R}_i, |\mathcal{R}_i| = r + 1$ for $i = 1, ..., \frac{n}{r+1}$
- Any set $S$ of $k$ symbols s.t. $\mathcal{R}_i \nsubseteq S$ suffices for decoding

Remarks:

- If $\exists i, \mathcal{R}_i \subseteq S, |S| = k$ then decoding of the codeword is impossible
An \((n, k, r)\) MR codes has the following properties:

- An \((n, k, r)\) LRC code
- Disjoint recovery sets \(\mathcal{R}_i, |\mathcal{R}_i| = r + 1\) for \(i = 1, \ldots, \frac{n}{r+1}\)
- Any set \(S\) of \(k\) symbols s.t. \(\mathcal{R}_i \not\subseteq S\) suffices for decoding

Remarks:

- If \(\exists i, \mathcal{R}_i \subseteq S, |S| = k\) then decoding of the codeword is impossible
- Terminology: MR codes = Partially MDS codes
$(n, k, r)$ MR codes - Definition

An $(n, k, r)$ MR codes has the following properties:

- An $(n, k, r)$ LRC code
- Disjoint recovery sets $\mathcal{R}_i$, $|\mathcal{R}_i| = r + 1$ for $i = 1, \ldots, \frac{n}{r+1}$
- Any set $S$ of $k$ symbols s.t. $\mathcal{R}_i \not\subseteq S$ suffices for decoding

Remarks:

- If $\exists i, \mathcal{R}_i \subseteq S$, $|S| = k$ then decoding of the codeword is impossible
- Terminology: MR codes = Partially MDS codes
- Equivalent definition using Matroid Theory language
Lemma

MR codes are **optimal** LRC codes
Lemma

MR codes are optimal LRC codes

Proof:

\[\text{Any } k + k r - 1 \text{ coordinates contain a subset } S \text{ s.t.}
\begin{align*}
1. |S| &= k \\
2. \forall i, R_i &\not\in S
\end{align*}
\]

By the MR property, \( S \) suffices for decoding.
MR codes - Basics

Lemma

*MR codes are optimal LRC codes*

Proof:

- Assume $r$ divides $k$

\[d = n - k - k + 2 \iff \text{any } d - 1 = n - k - k + 1 \text{ erasures are recoverable} \iff \text{any } n - (d - 1) = k + k - 1 \text{ coordinates suffice for decoding}

- Any $k + k - 1$ coordinates contain a subset $S$ such that
  1. $|S| = k$
  2. $\forall i, R_i \not\in S$

  By the MR property, $S$ suffices for decoding.
Lemma

**MR codes are optimal LRC codes**

Proof:

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

*MR codes are optimal LRC codes*

Proof:

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

\[ d = n - k - \frac{k}{r} + 2 \]
Lemma

*MR codes are optimal LRC codes*

Proof:

- Assume $r$ divides $k$

- MR code is an Optimal LRC code

\[ d = n - k - \frac{k}{r} + 2 \]

\[ \Leftrightarrow \text{any } d - 1 = n - k - \frac{k}{r} + 1 \text{ erasures are recoverable} \]
Lemma

MR codes are optimal LRC codes

Proof:

- Assume $r$ divides $k$

- MR code is an Optimal LRC code

\[ \iff d = n - k - \frac{k}{r} + 2 \]

\[ \iff \text{any } d - 1 = n - k - \frac{k}{r} + 1 \text{ erasures are recoverable} \]

\[ \iff \text{any } n - (d - 1) = k + \frac{k}{r} - 1 \text{ coordinates suffice for decoding} \]
Lemma

*MR codes are optimal LRC codes*

Proof:

- Assume $r$ divides $k$

- MR code is an Optimal LRC code

\[ d = n - k - \frac{k}{r} + 2 \]

\[ \iff \text{any } d - 1 = n - k - \frac{k}{r} + 1 \text{ erasures are recoverable} \]

\[ \iff \text{any } n - (d - 1) = k + \frac{k}{r} - 1 \text{ coordinates suffice for decoding} \]

- Any $k + \frac{k}{r} - 1$ coordinates contain a subset $S$ s.t.
MR codes - Basics

Lemma

MR codes are optimal LRC codes

Proof:

- Assume $r$ divides $k$
- MR code is an Optimal LRC code
  
  \[ d = n - k - \frac{k}{r} + 2 \]
  
  \[ \iff \text{any } d - 1 = n - k - \frac{k}{r} + 1 \text{ erasures are recoverable} \]
  
  \[ \iff \text{any } n - (d - 1) = k + \frac{k}{r} - 1 \text{ coordinates suffice for decoding} \]
- Any $k + \frac{k}{r} - 1$ coordinates contain a subset $S$ s.t.
  1. $|S| = k$
Mr codes - Basics

Lemma

MR codes are **optimal** LRC codes

Proof:

- Assume \( r \) divides \( k \)

- MR code is an Optimal LRC code

  \[ \iff d = n - k - \frac{k}{r} + 2 \]

  \[ \iff \text{any } d - 1 = n - k - \frac{k}{r} + 1 \text{ erasures are recoverable} \]

  \[ \iff \text{any } n - (d - 1) = k + \frac{k}{r} - 1 \text{ coordinates suffice for decoding} \]

- Any \( k + \frac{k}{r} - 1 \) coordinates contain a subset \( S \) s.t.
  1. \( |S| = k \)
  2. \( \forall i, R_i \not\in S \)
Lemma

MR codes are optimal LRC codes

Proof:

- Assume $r$ divides $k$

- MR code is an Optimal LRC code
  \[ d = n - k - \frac{k}{r} + 2 \]
  \[ \iff \text{any } d - 1 = n - k - \frac{k}{r} + 1 \text{ erasures are recoverable} \]
  \[ \iff \text{any } n - (d - 1) = k + \frac{k}{r} - 1 \text{ coordinates suffice for decoding} \]

- Any $k + \frac{k}{r} - 1$ coordinates contain a subset $S$ s.t.
  1. $|S| = k$
  2. $\forall i, R_i \not\subseteq S$

- By the MR property, $S$ suffices for decoding
MR codes - Basics

Lemma

*MR codes are optimal LRC codes*
Lemma

MR codes are optimal LRC codes
Lemma

MR codes are optimal LRC codes
Lemma

*MR codes are optimal LRC codes*
Lemma

*MR codes are optimal LRC codes*

Q: MR codes = Optimal LRC codes ?
Lemma

*MR codes are optimal LRC codes*

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

- Probabilistic construction of optimal LRC code results an MR code
  1. Non explicit
  2. Field size is superpolynomial in the length

- Explicit constructions
  - Any \( n, k, r \)
  - Field size is superpolynomial

- Construction of Partially MDS codes
  - Restricted parameters
  - Field size \( |F| = O(n) \)
Constructing MR codes

• Probabilistic construction of optimal LRC code results an MR code
Constructing MR codes

• Probabilistic construction of optimal LRC code results an MR code
  1. Non explicit
Constructing MR codes

- Probabilistic construction of optimal LRC code results an MR code
  1. Non explicit
  2. Field size is superpolynomial in the length (is this truly necessary?)
Constructing MR codes

- Probabilistic construction of optimal LRC code results an MR code
  1. Non explicit
  2. Field size is superpolynomial in the length (is this truly necessary?)

- Explicit constructions [Rawat, Koyluoglu, Silberstein, Vishwanath 14, Gopalan, Huang, Jenkins, Yekhanin 14, Tamo, Papailiopoulos, Dimakis 14]
Constructing MR codes

- Probabilistic construction of optimal LRC code results an MR code
  1. Non explicit
  2. Field size is superpolynomial in the length (is this truly necessary?)

- Explicit constructions [Rawat, Koyluoglu, Silberstein, Vishwanath 14, Gopalan, Huang, Jenkins, Yekhanin 14, Tamo, Papailiopoulos, Dimakis 14]
  1. Any $n, k, r$
Constructing MR codes

- Probabilistic construction of optimal LRC code results an MR code
  1. Non explicit
  2. Field size is superpolynomial in the length (is this truly necessary?)

- Explicit constructions [Rawat, Koyluoglu, Silberstein, Vishwanath 14, Gopalan, Huang, Jenkins, Yekhanin 14, Tamo, Papailiopoulos, Dimakis 14]
  1. Any $n, k, r$
  2. Field size is superpolynomial
Constructing MR codes

- Probabilistic construction of optimal LRC code results an MR code
  1. Non explicit
  2. Field size is superpolynomial in the length (is this truly necessary?)

- Explicit constructions [Rawat, Koyluoglu, Silberstein, Vishwanath 14, Gopalan, Huang, Jenkins, Yekhanin 14, Tamo, Papailiopoulos, Dimakis 14]
  1. Any $n, k, r$
  2. Field size is superpolynomial

- Construction of Partially MDS codes [Blaum, Lee Hafner, Hetzler 13, Blaum, Plank, Schwartz, Yaakobi 14]
Constructing MR codes

- Probabilistic construction of optimal LRC code results an MR code
  1. Non explicit
  2. Field size is superpolynomial in the length (is this truly necessary?)

- Explicit constructions [Rawat, Koyluoglu, Silberstein, Vishwanath 14, Gopalan, Huang, Jenkins, Yekhanin 14, Tamo, Papailiopoulos, Dimakis 14]
  1. Any $n, k, r$
  2. Field size is superpolynomial

- Construction of Partially MDS codes [Blaum, Lee Hafner, Hetzler 13, Blaum, Plank, Schwartz, Yaakobi 14]
  1. Restricted parameters
Constructing MR codes

• Probabilistic construction of optimal LRC code results an MR code
  1. Non explicit
  2. Field size is superpolynomial in the length (is this truly necessary?)

• Explicit constructions [Rawat, Koyluoglu, Silberstein, Vishwanath 14, Gopalan, Huang, Jenkins, Yekhanin 14, Tamo, Papailiopoulos, Dimakis 14]
  1. Any \( n, k, r \)
  2. Field size is superpolynomial

• Construction of Partially MDS codes [Blaum, Lee Hafner, Hetzler 13, Blaum, Plank, Schwartz, Yaakobi 14]
  1. Restricted parameters
  2. Field size \( |\mathbb{F}| = O(n) \)
Constructing MR codes - Easy cases

1. An \((n,k)\) RS is an \((n,k,k)\) MR code.

2. \(|F| = O(n)\).

Q: Is the RS-like construction an MR code?

1. No, in general
2. Yes in some cases (other cases?)
Constructing MR codes - Easy cases

• $r = k$
Constructing MR codes - Easy cases

- $r = k$
  
  1. An $(n,k)$ RS is an $(n,k,k)$ MR code
Constructing MR codes - Easy cases

- $r = k$
  1. An $(n,k)$ RS is an $(n,k,k)$ MR code
  2. $|F| = O(n)$
Constructing MR codes - Easy cases

- $r = k$
  - 1. An $(n, k)$ RS is an $(n, k, k)$ MR code
  - 2. $|\mathbb{F}| = O(n)$

- $r = 1$
Constructing MR codes - Easy cases

- \( r = k \)
  1. An \((n, k)\) RS is an \((n, k, k)\) MR code
  2. \(|\mathbb{F}| = O(n)\)

- \( r = 1 \)
  1. Duplication of an \((n/2, k)\) RS is an \((n, k, 1)\) MR code
Constructing MR codes - Easy cases

- \( r = k \)
  1. An \((n, k)\) RS is an \((n, k, k)\) MR code
  2. \(|\mathbb{F}| = O(n)\)

- \( r = 1 \)
  1. Duplication of an \((n/2, k)\) RS is an \((n, k, 1)\) MR code
  2. \(|\mathbb{F}| = O(n)\)
Constructing MR codes - Easy cases

• $r = k$
  1. An $(n, k)$ RS is an $(n, k, k)$ MR code
  2. $|\mathbb{F}| = O(n)$

• $r = 1$
  1. Duplication of an $(n/2, k)$ RS is an $(n, k, 1)$ MR code
  2. $|\mathbb{F}| = O(n)$

• Q: Is the RS-like construction an MR code?
Constructing MR codes - Easy cases

- $r = k$
  1. An $(n, k)$ RS is an $(n, k, k)$ MR code
  2. $|\mathbb{F}| = O(n)$

- $r = 1$
  1. Duplication of an $(n/2, k)$ RS is an $(n, k, 1)$ MR code
  2. $|\mathbb{F}| = O(n)$

- Q: Is the RS-like construction an MR code?
  1. No, in general
Constructing MR codes - Easy cases

- $r = k$
  1. An $(n, k)$ RS is an $(n, k, k)$ MR code
  2. $|\mathbb{F}| = O(n)$

- $r = 1$
  1. Duplication of an $(n/2, k)$ RS is an $(n, k, 1)$ MR code
  2. $|\mathbb{F}| = O(n)$

- Q: Is the RS-like construction an MR code?
  1. No, in general
  2. Yes in some cases (other cases?)
Basic facts:

- $F_q^m$ is a vector space of dimension $m$ over $F_q$.
- Linearized polynomial has the form $f(x) = \sum_{i=0}^{m-1} a_i x^{q^i}$, where $a_i \in F_q^m$.
- For $x, y \in F_q^m$ and $\alpha_1, \alpha_2 \in F_q$, $f(\alpha_1 x + \alpha_2 y) = \alpha_1 f(x) + \alpha_2 f(y)$. 
MR codes through linearized polynomials

Basic facts:

- $\mathbb{F}_{q^m}$ is a vector space of dimension $m$ over $\mathbb{F}_q$
MR codes through linearized polynomials

Basic facts:

- $\mathbb{F}_{q^m}$ is a vector space of dimension $m$ over $\mathbb{F}_q$

- Linearized polynomial has the form

$$f(x) = \sum_{i=0}^{m-1} a_i x^{q^i}, \text{ where } a_i \in \mathbb{F}_{q^m}$$
Basic facts:

- $\mathbb{F}_{q^m}$ is a vector space of dimension $m$ over $\mathbb{F}_q$.

- Linearized polynomial has the form

$$f(x) = \sum_{i=0}^{m-1} a_i x^{q^i}, \text{ where } a_i \in \mathbb{F}_{q^m}$$

- For $x, y \in \mathbb{F}_{q^m}$ and $\alpha_1, \alpha_2 \in \mathbb{F}_q$

$$f(\alpha_1 x + \alpha_2 y) = \alpha_1 f(x) + \alpha_2 f(y)$$
Constructing an \((n, k, r)\) MR code [Rawat, Koyluoglu, Silberstein, Vishwanath 14]

- Set \(m = \frac{nr}{r+1}\)
Constructing an \((n, k, r)\) MR code [Rawat, Koyluoglu, Silberstein, Vishwanath 14]

- Set \(m = \frac{nr}{r+1}\)
- \(\alpha_1, \ldots, \alpha_m \in \mathbb{F}_2^m\) linearly independent elements over \(\mathbb{F}_2\)
Constructing an \((n, k, r)\) MR code [Rawat, Koyluoglu, Silberstein, Vishwanath 14]

- Set \(m = \frac{nr}{r+1}\)

- \(\alpha_1, \ldots, \alpha_m \in \mathbb{F}_{2^m}\) linearly independent elements over \(\mathbb{F}_2\)

- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^{r} \alpha_i\}\)
Constructing an \((n, k, r)\) MR code [Rawat, Koyluoglu, Silberstein, Vishwanath 14]

- Set \(m = \frac{nr}{r+1}\)
- \(\alpha_1, \ldots, \alpha_m \in \mathbb{F}_{2^m}\) linearly independent elements over \(\mathbb{F}_2\)
- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^r \alpha_i\}, \mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}\)
Constructing an \((n, k, r)\) MR code [Rawat, Koyluoglu, Silberstein, Vishwanath 14]

- Set \(m = \frac{nr}{r+1}\)

- \(\alpha_1, \ldots, \alpha_m \in \mathbb{F}_{2^m}\) linearly independent elements over \(\mathbb{F}_2\)

- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^{r} \alpha_i\}, \quad \mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}, \ldots, \mathcal{R}_{\frac{n}{r+1}}\)
Constructing an \((n, k, r)\) MR code \([\text{Rawat, Koyluoglu, Silberstein, Vishwanath 14}]\)

- Set \(m = \frac{nr}{r+1}\)

- \(\alpha_1, \ldots, \alpha_m \in \mathbb{F}_{2^m}\) linearly independent elements over \(\mathbb{F}_2\)

- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^{r} \alpha_i\}, \mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}, \ldots, \mathcal{R}_{\frac{n}{r+1}}\)

- Observations:
Constructing an \((n, k, r)\) MR code [Rawat, Koyluoglu, Silberstein, Vishwanath 14]

- Set \(m = \frac{nr}{r+1}\)

- \(\alpha_1, \ldots, \alpha_m \in \mathbb{F}_2^m\) linearly independent elements over \(\mathbb{F}_2\)

- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^{r} \alpha_i\}, \mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}, \ldots, \mathcal{R}_{n/(r+1)}\)

- Observations:
  1. \(|\bigcup_i \mathcal{R}_i| = n\)
Constructing an \((n, k, r)\) MR code [Rawat, Koyluoglu, Silberstein, Vishwanath 14]

- Set \(m = \frac{nr}{r+1}\)
- \(\alpha_1, \ldots, \alpha_m \in \mathbb{F}_{2^m}\) linearly independent elements over \(\mathbb{F}_2\)
- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^r \alpha_i\}, \mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}, \ldots, \mathcal{R}_{n(r+1)}\)
- Observations:
  1. \(|\cup_i \mathcal{R}_i| = n\)
  2. \(\sum_{\alpha \in \mathcal{R}_i} \alpha = 0\)
Constructing an \((n, k, r)\) MR code [Rawat, Koyluoglu, Silberstein, Vishwanath 14]

- Set \(m = \frac{nr}{r+1}\)

- \(\alpha_1, \ldots, \alpha_m \in \mathbb{F}_2^m\) linearly independent elements over \(\mathbb{F}_2\)

- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^r \alpha_i\}\), \(\mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}\), \ldots, \(\mathcal{R}_{\frac{n}{r+1}}\)

- Observations:
  1. \(|\bigcup_i \mathcal{R}_i| = n|
  2. \(\sum_{\alpha \in \mathcal{R}_i} \alpha = 0\)

Encoding:
Constructing an \((n, k, r)\) MR code [Rawat, Koyluoglu, Silberstein, Vishwanath 14]

- Set \(m = \frac{nr}{r+1}\)
- \(\alpha_1, \ldots, \alpha_m \in \mathbb{F}_2^m\) linearly independent elements over \(\mathbb{F}_2\)
- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^{r} \alpha_i\}, \mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}, \ldots, \mathcal{R}_{\frac{n}{r+1}}\)

**Observations:**

1. \(|\bigcup_i \mathcal{R}_i| = n\)
2. \(\sum_{\alpha \in \mathcal{R}_i} \alpha = 0\)

**Encoding:** \((v_0, \ldots, v_{k-1})\)

1. Given \(k\) information symbols of \(\mathbb{F}_2^m\)
Constructing an \((n, k, r)\) MR code [Rawat, Koyluoglu, Silberstein, Vishwanath 14]

- Set \(m = \frac{nr}{r+1}\)

- \(\alpha_1, \ldots, \alpha_m \in \mathbb{F}_{2^m}\) linearly independent elements over \(\mathbb{F}_2\)

- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^r \alpha_i\}, \mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}, \ldots, \mathcal{R}_{\frac{n}{r+1}}\)

- Observations:
  1. \(|\bigcup_i \mathcal{R}_i| = n\)
  2. \(\sum_{\alpha \in \mathcal{R}_i} \alpha = 0\)

Encoding: \((v_0, \ldots, v_{k-1}) \mapsto f(x) = \sum_{i=0}^{k-1} v_i x^{2^i}\)

1. Given \(k\) information symbols of \(\mathbb{F}_{2^m}\)

2. Form the linearized polynomial
Constructing an \((n, k, r)\) MR code [Rawat, Koyluoglu, Silberstein, Vishwanath 14]

- Set \(m = \frac{nr}{r+1}\)

- \(\alpha_1, \ldots, \alpha_m \in \mathbb{F}_{2^m}\) linearly independent elements over \(\mathbb{F}_2\)

- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^{r} \alpha_i\}, \mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}, \ldots, \mathcal{R}_{n-r+1}\)

- Observations:
  1. \(|\bigcup_i \mathcal{R}_i| = n\)
  2. \(\sum_{\alpha \in \mathcal{R}_i} \alpha = 0\)

**Encoding:** \((v_0, \ldots, v_{k-1}) \mapsto f(x) = \sum_{i=0}^{k-1} v_i x^{2^i} \mapsto (f(\alpha) : \alpha \in \bigcup_i \mathcal{R}_i)\)

1. Given \(k\) information symbols of \(\mathbb{F}_{2^m}\)
2. Form the linearized polynomial
3. Output the length\(–n\) codeword
Constructing an \((n, k, r)\) MR code - Cont’d

Locality: Recover \(f(\beta)\) for \(\beta \in \mathbb{R}\)

\[\sum_{\alpha \in \mathbb{R}} \alpha = 0 \Rightarrow \beta = \sum_{\alpha \in \mathbb{R}} \beta \alpha \Rightarrow f(\beta) = f(\sum_{\alpha \in \mathbb{R}} \beta \alpha) = \sum_{\alpha \in \mathbb{R}} \beta f(\alpha)\]

MR Property:

• \(= \Rightarrow 2^k\) evaluations of \(f(x)\):

\[f(\sum_{\epsilon s \in \{0, 1\}} s \epsilon s) = \sum_{\epsilon s \in \{0, 1\}} s \epsilon s f(s)\]

• \(\deg(f) \leq 2^k - 1\), since \(f(x) = \sum_{i=0}^{k-1} v_i x^{2^i}\) can be interpolated
Locality: Recover \( f(\beta) =? \) for \( \beta \in R_i \)
Constructing an \((n, k, r)\) MR code - Cont’d

**Locality:** Recover \(f(\beta) = \, ? \) for \(\beta \in \mathcal{R}_i\)

\[
\sum_{\alpha \in \mathcal{R}_i} \alpha = 0
\]
Locality: Recover $f(\beta) = ?$ for $\beta \in \mathcal{R}_i$

\[
\sum_{\alpha \in \mathcal{R}_i} \alpha = 0 \implies \beta = \sum_{\alpha \in \mathcal{R}_i \setminus \beta} \alpha
\]
Locality: Recover $f(\beta) =$? for $\beta \in \mathcal{R}_i$

$$\sum_{\alpha \in \mathcal{R}_i} \alpha = 0 \implies \beta = \sum_{\alpha \in \mathcal{R}_i \setminus \beta} \alpha \implies f(\beta) = f\left( \sum_{\alpha \in \mathcal{R}_i \setminus \beta} \alpha \right)$$
Locality: Recover $f(\beta) = ?$ for $\beta \in \mathcal{R}_i$

$$\sum_{\alpha \in \mathcal{R}_i} \alpha = 0 \implies \beta = \sum_{\alpha \in \mathcal{R}_i \setminus \beta} \alpha \implies f(\beta) = f(\sum_{\alpha \in \mathcal{R}_i \setminus \beta} \alpha) = \sum_{\alpha \in \mathcal{R}_i \setminus \beta} f(\alpha)$$
Constructing an \((n, k, r)\) MR code - Cont’d

**MR Property:**

\[
\text{deg}(f) \leq 2k - 1,
\] since

\[
f(x) = \sum_{i=0}^{k-1} v_i x^{2i},
\] can be interpolated.
Constructing an \((n, k, r)\) MR code - Cont’d

MR Property:

- \(\alpha_1, \ldots, \alpha_m\) are linearly independent over \(\mathbb{F}_2\)
Constructing an \((n, k, r)\) MR code - Cont’d

**MR Property:**

- \(\alpha_1, ..., \alpha_m\) are linearly independent over \(\mathbb{F}_2\)
- \(\mathcal{R}_1 = \{\alpha_1, ..., \alpha_r, \sum_{i=1}^r \alpha_i\}\)
Constructing an \((n, k, r)\) MR code - Cont’d

**MR Property:**

- \(\alpha_1, \ldots, \alpha_m\) are linearly independent over \(\mathbb{F}_2\)

- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^{r} \alpha_i\}\), \(\mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}\)
Constructing an \((n, k, r)\) MR code - Cont’d

MR Property:

- \(\alpha_1, \ldots, \alpha_m\) are linearly independent over \(\mathbb{F}_2\)
- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^{r} \alpha_i\}, \mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}, \ldots, \mathcal{R}_{\frac{n}{r+1}}\)
Constructing an \((n, k, r)\) MR code - Cont’d

**MR Property:**

- \(\alpha_1, ..., \alpha_m\) are linearly independent over \(\mathbb{F}_2\)

- \(\mathcal{R}_1 = \{\alpha_1, ..., \alpha_r, \sum_{i=1}^{r} \alpha_i\},\ \mathcal{R}_2 = \{\alpha_{r+1}, ..., \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}, \ldots, \mathcal{R}_{\frac{n}{r+1}}\)

- \(\implies\) the only linear dependencies in \(\bigcup_i \mathcal{R}_i\) are the \(\mathcal{R}_i\)'s
Constructing an $(n, k, r)$ MR code - Cont’d

**MR Property:**

- $\alpha_1, \ldots, \alpha_m$ are linearly independent over $\mathbb{F}_2$
- $\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^{r} \alpha_i\}$, $\mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}$, $\ldots$, $\mathcal{R}_{n-r+1}$

$\implies$ the only linear dependencies in $\bigcup_i \mathcal{R}_i$ are the $\mathcal{R}_i$’s

$\implies$ If $S \subseteq \bigcup_i \mathcal{R}_i$ s.t. $|S| = k$, $\mathcal{R}_i \notin S$ then the elements of $S$ are linearly independent over $\mathbb{F}_2$
Constructing an \((n, k, r)\) MR code - Cont’d

**MR Property:**

- \(\alpha_1, ..., \alpha_m\) are linearly independent over \(F_2\)

- \(R_1 = \{\alpha_1, ..., \alpha_r, \sum_{i=1}^{r} \alpha_i\}, R_2 = \{\alpha_{r+1}, ..., \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}, ..., R_{n-r+1}\)

- \(\Rightarrow\) the only linear dependencies in \(\cup_i R_i\) are the \(R_i\)’s

- \(\Rightarrow\) If \(S \subseteq \cup_i R_i\) s.t. \(|S| = k, R_i \not\subset S\) then the elements of \(S\) are linearly independent over \(F_2\)

- \(\Rightarrow\) all \(2^k\) possible sums \(\sum_{s \in S} \epsilon_s s\) are distinct
Constructing an \((n, k, r)\) MR code - Cont’d

**MR Property:**

- \(\alpha_1, \ldots, \alpha_m\) are linearly independent over \(\mathbb{F}_2\)

- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^{r} \alpha_i\}, \mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}, \ldots, \mathcal{R}_{\frac{n}{r+1}}\)

- \(\implies\) the only linear dependencies in \(\bigcup_i \mathcal{R}_i\) are the \(\mathcal{R}_i\)’s

- \(\implies\) If \(S \subseteq \bigcup_i \mathcal{R}_i\) s.t. \(|S| = k, \mathcal{R}_i \not\subseteq S\) then the elements of \(S\) are linearly independent over \(\mathbb{F}_2\)

- \(\implies\) all \(2^k\) possible sums \(\sum_{s \in S} \epsilon_s \in \{0, 1\} \epsilon_s s\) are distinct

- \(\implies\) \(2^k\) evaluations of \(f(x)\):
Constructing an \((n, k, r)\) MR code - Cont’d

**MR Property:**

- \(\alpha_1, \ldots, \alpha_m\) are linearly independent over \(\mathbb{F}_2\)
- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^{r} \alpha_i\}, \mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}, \ldots, \mathcal{R}_{r+1}\)

\(\implies\) the only linear dependencies in \(\bigcup_i \mathcal{R}_i\) are the \(\mathcal{R}_i\)’s

\(\implies\) If \(S \subseteq \bigcup_i \mathcal{R}_i\) s.t. \(|S| = k, \mathcal{R}_i \not\subseteq S\) then the elements of \(S\) are linearly independent over \(\mathbb{F}_2\)

\(\implies\) all \(2^k\) possible sums \(\sum_{\epsilon_s \in \{0,1\}} \epsilon_s s\) are distinct

\(\implies\) \(2^k\) evaluations of \(f(x)\):

\[f\left(\sum_{\epsilon_s \in \{0,1\}} \epsilon_s s\right)\]
Constructing an \((n, k, r)\) MR code - Cont’d

**MR Property:**

- \(\alpha_1, \ldots, \alpha_m\) are linearly independent over \(\mathbb{F}_2\)
- \(\mathcal{R}_1 = \{\alpha_1, \ldots, \alpha_r, \sum_{i=1}^r \alpha_i\}\), \(\mathcal{R}_2 = \{\alpha_{r+1}, \ldots, \alpha_{2r}, \sum_{i=r+1}^{2r} \alpha_i\}\), \(\ldots, \mathcal{R}_{n-r+1}\)

- \(\implies\) the only linear dependencies in \(\cup_i \mathcal{R}_i\) are the \(\mathcal{R}_i\)’s
- \(\implies\) If \(S \subseteq \cup_i \mathcal{R}_i\) s.t. \(|S| = k\), \(\mathcal{R}_i \not\subseteq S\) then the elements of \(S\) are linearly independent over \(\mathbb{F}_2\)
- \(\implies\) all \(2^k\) possible sums \(\sum_{\substack{\epsilon_s \in \{0,1\} \ s \in S}} \epsilon_s s\) are distinct
- \(\implies\) \(2^k\) evaluations of \(f(x)\):

\[
\begin{align*}
f\left(\sum_{\substack{\epsilon_s \in \{0,1\} \ s \in S}} \epsilon_s s\right) &= \sum_{\epsilon_s \in \{0,1\}} \epsilon_s f(s)
\end{align*}
\]
Constructing an \((n, k, r)\) MR code - Cont’d

**MR Property:**

- \[ \Rightarrow \quad 2^k \text{ evaluations of } f(x): \]

\[
f\left( \sum_{\epsilon_s \in \{0, 1\}} \epsilon_s s \right) = \sum_{\epsilon_s \in \{0, 1\}} \epsilon_s f(s)
\]
Constructing an \((n, k, r)\) MR code - Cont'd

MR Property:

- \(\implies 2^k\) evaluations of \(f(x)\):

\[
f \left( \sum_{\epsilon_s \in \{0, 1\}} \epsilon_s s \right) = \sum_{\epsilon_s \in \{0, 1\}} \epsilon_s f(s)
\]

- \(\deg(f) \leq 2^{k-1}\)
Constructing an \( (n, k, r) \) MR code - Cont’d

**MR Property:**

- \( \implies 2^k \) evaluations of \( f(x) \):

\[
  f \left( \sum_{s \in S} \epsilon_s s \right) = \sum_{s \in S} \epsilon_s f(s)
\]

- \( \deg(f) \leq 2^{k-1} \), since \( f(x) = \sum_{i=0}^{k-1} v_i x^{2^i} \)
Constructing an \((n, k, r)\) MR code - Cont’d

MR Property:

- \(\implies 2^k\) evaluations of \(f(x)\):

\[
f \left( \sum_{\epsilon_s \in \{0, 1\}} \epsilon_s s \right) = \sum_{\epsilon_s \in \{0, 1\}} \epsilon_s f(s)
\]

- \(\deg(f) \leq 2^{k-1}\), since \(f(x) = \sum_{i=0}^{k-1} v_i x^i\)

- \(f(x)\) can be interpolated
Properties of the MR code construction

• Flexible set of parameters $n$, $k$, $r$

• Need $m = nr + 1$ linearly independent elements over $\mathbb{F}_2$ $\Rightarrow |\mathbb{F}_2| = 2^{nr+1}$

• Field size is exponential in $n \times k$
Properties of the MR code construction

- Flexible set of parameters $n, k, r$
Properties of the MR code construction

- Flexible set of parameters $n, k, r$ ✓

Field size is exponential in $n$
Properties of the MR code construction

- Flexible set of parameters $n, k, r$

- Need $m = \frac{nr}{r+1}$ linearly independent elements over $\mathbb{F}_2$
Properties of the MR code construction

- Flexible set of parameters $n, k, r$

- Need $m = \frac{nr}{r+1}$ linearly independent elements over $\mathbb{F}_2 \implies |\mathbb{F}| = 2^{\frac{nr}{r+1}}$
Properties of the MR code construction

- 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 $n$
Properties of the MR code construction

- 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 $n$ ❌
Properties of the MR code construction

- Flexible set of parameters $n, k, r \checkmark$

- 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 $n \times$

- Can we do better?
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

• In any \((n, k, r)\) LRC code, \(k \leq nr + 1\)

• High rate means \(nr + 1 - k = O(n \log n)\)

• \(F(n, k, r)\) is the minimum field size needed for an \((n, k, r)\) MR code.

• Known bounds on \(F(n, k, r)\):
  \[
  nr + 1 - k \geq 5
  \]
  \[
  F(n, k, r) \leq 2r + 1 + \theta(n)O(n^{3/2})O(n^{7/3})O(nr + 1 - k - 1)\]
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

• In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

- In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)

- High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

- In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)
- High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)
- \(F(n, k, r)\) = minimum field size needed for an \((n, k, r)\) MR code
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

- In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)

- High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)

- \(\mathbb{F}(n, k, r) = \) minimum field size needed for an \((n, k, r)\) MR code

- Known bounds on \(\mathbb{F}(n, k, r)\):
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

- In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)

- High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)

- \(\mathbb{F}(n, k, r) = \) minimum field size needed for an \((n, k, r)\) MR code

- Known bounds on \(\mathbb{F}(n, k, r)\):

\[
\begin{array}{c|c|c|c|c|c}
\frac{nr}{r+1} - k & \mathbb{F}(n, k, r) \\
\hline
\end{array}
\]
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

- In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)

- High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)

- \(\mathbb{F}(n, k, r)\) = minimum field size needed for an \((n, k, r)\) MR code

- Known bounds on \(\mathbb{F}(n, k, r)\):

  \[
  \begin{array}{c|c|c|c|c}
  \frac{nr}{r+1} - k & 0 & & & \\
  \mathbb{F}(n, k, r) & & & & \\
  \end{array}
  \]
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

- In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)

- High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)

- \(\mathbb{F}(n, k, r) = \) minimum field size needed for an \((n, k, r)\) MR code

- Known bounds on \(\mathbb{F}(n, k, r)\):

<table>
<thead>
<tr>
<th>(\frac{nr}{r+1} - k)</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\mathbb{F}(n, k, r))</td>
<td>2</td>
</tr>
</tbody>
</table>

The code is a \(\frac{n}{r+1}\) independent parity check codes of length \(r + 1\)
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

• In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)

• High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)

• \(\mathbb{F}(n, k, r)\) = minimum field size needed for an \((n, k, r)\) MR code

• Known bounds on \(\mathbb{F}(n, k, r)\):

\[
\begin{array}{c|c|c|c|c|c}
\frac{nr}{r+1} - k & 0 & 1 & \cdots & 2 & \cdots \\
\mathbb{F}(n, k, r) & 2 & 1 & \cdots & 0 & \cdots
\end{array}
\]
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

- In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)

- High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)

- \(F(n, k, r)\) = minimum field size needed for an \((n, k, r)\) MR code

- Known bounds on \(F(n, k, r)\):

<table>
<thead>
<tr>
<th>(\frac{nr}{r+1} - k)</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(F(n, k, r))</td>
<td>2</td>
<td>(r + 1)</td>
</tr>
</tbody>
</table>
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

- In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)

- High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)

- \(\mathbb{F}(n, k, r) = \) minimum field size needed for an \((n, k, r)\) MR code

- Known bounds on \(\mathbb{F}(n, k, r)\):

\[
\begin{array}{c|c|c|c}
\frac{nr}{r+1} - k & 0 & 1 & 2 \\
\hline
\mathbb{F}(n, k, r) & 2 & r + 1 & \\
\end{array}
\]
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

- In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)

- High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)

- \(\mathbb{F}(n, k, r)\) = minimum field size needed for an \((n, k, r)\) MR code

- Known bounds on \(\mathbb{F}(n, k, r)\):

\[
\begin{array}{c|c|c|c|}
\frac{nr}{r+1} - k & 0 & 1 & 2 \\
\hline
\mathbb{F}(n, k, r) & 2 & r + 1 & \theta(n)
\end{array}
\]
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

- In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)

- High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)

- \(\mathbb{F}(n, k, r)\) = minimum field size needed for an \((n, k, r)\) MR code

- Known bounds on \(\mathbb{F}(n, k, r)\):

<table>
<thead>
<tr>
<th>(\frac{nr}{r+1} - k)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3[3/2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\mathbb{F}(n, k, r))</td>
<td>2</td>
<td>(r + 1)</td>
<td>(\theta(n))</td>
<td>(O(n^{3/2}))</td>
</tr>
</tbody>
</table>
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

- In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)

- High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)

- \(\mathbb{F}(n, k, r) = \) minimum field size needed for an \((n, k, r)\) MR code

- Known bounds on \(\mathbb{F}(n, k, r)\):

<table>
<thead>
<tr>
<th>(\frac{nr}{r+1} - k)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\mathbb{F}(n, k, r))</td>
<td>2</td>
<td>(r + 1)</td>
<td>(\theta(n))</td>
<td>(O(n^{3/2}))</td>
</tr>
</tbody>
</table>
In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)

High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)

\(F(n, k, r) = \text{minimum field size needed for an } (n, k, r) \text{ MR code}\)

Known bounds on \(F(n, k, r)\):

<table>
<thead>
<tr>
<th>(\frac{nr}{r+1} - k)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(F(n, k, r))</td>
<td>2</td>
<td>(r + 1)</td>
<td>(\theta(n))</td>
<td>(O(n^{3/2}))</td>
<td></td>
</tr>
</tbody>
</table>
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

- In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)

- High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)

- \(\mathbb{F}(n, k, r) = \text{minimum field size needed for an } (n, k, r) \text{ MR code}\)

- Known bounds on \(\mathbb{F}(n, k, r)\):

<table>
<thead>
<tr>
<th>(\frac{nr}{r+1} - k)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\mathbb{F}(n, k, r))</td>
<td>2</td>
<td>(r + 1)</td>
<td>(\theta(n))</td>
<td>(O(n^{3/2}))</td>
<td>(O(n^{7/3}))</td>
</tr>
</tbody>
</table>
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

- In any $(n, k, r)$ LRC code $k \leq \frac{nr}{r+1}$

- High rate means $\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)$

- $F(n, k, r) =$ minimum field size needed for an $(n, k, r)$ MR code

- Known bounds on $F(n, k, r)$:

\[
\begin{array}{c|c|c|c|c|c|c}
\frac{nr}{r+1} - k & 0 & 1 & 2 & 3 & 4 & \geq 5 \\
F(n, k, r) & 2 & r + 1 & \theta(n) & O(n^{3/2}) & O(n^{7/3}) & \\
\end{array}
\]
High-rate MR codes [Gopalan, Huang, Jenkins, Yekhanin 14]

- In any \((n, k, r)\) LRC code \(k \leq \frac{nr}{r+1}\)
- High rate means \(\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)\)
- \(\mathbb{F}(n, k, r) = \text{minimum field size needed for an } (n, k, r) \text{ MR code}\)
- Known bounds on \(\mathbb{F}(n, k, r)\):

\[
\begin{array}{ccccccc}
\frac{nr}{r+1} - k & 0 & 1 & 2 & 3 & 4 & \geq 5 \\
\mathbb{F}(n, k, r) & 2 & r + 1 & \theta(n) & O(n^{3/2}) & O(n^{7/3}) & O(n^{\frac{nr}{r+1} - k - 1})
\end{array}
\]
Summary of best known constructions of MR codes

<table>
<thead>
<tr>
<th></th>
<th>High-Rate</th>
<th>Constant-Rate</th>
<th>Vanishing Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field size</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field size

\[ n \leq \frac{nr}{r+1-k} = O(n \log n) \]

High-Rate Regime: Gopalan, Huang, Jenkins, Yekhanin 14

Constant-Rate Regime: Rawat, Koyluoglu, Silberstein, Vishwanath 14

Vanishing Rate Regime: T., Papailiopoulos, Dimakis 14
### Summary of best known constructions of MR codes

<table>
<thead>
<tr>
<th>Dimension</th>
<th>High-Rate</th>
<th>Constant-Rate</th>
<th>Vanishing Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field size</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **High-Rate Regime**: Gopalan, Huang, Jenkins, Yekhanin 14
Summary of best known constructions of MR codes

<table>
<thead>
<tr>
<th></th>
<th>High-Rate</th>
<th>Constant-Rate</th>
<th>Vanishing Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimension</strong></td>
<td>( \frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Field size</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **High-Rate Regime**: Gopalan, Huang, Jenkins, Yekhanin 14
Summary of best known constructions of MR codes

<table>
<thead>
<tr>
<th>Dimension</th>
<th>High-Rate</th>
<th>Constant-Rate</th>
<th>Vanishing Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field size</td>
<td>( n^{\frac{nr}{r+1} - k - 1} )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **High-Rate Regime:** Gopalan, Huang, Jenkins, Yekhanin 14
Summary of best known constructions of MR codes

<table>
<thead>
<tr>
<th></th>
<th>High-Rate</th>
<th>Constant-Rate</th>
<th>Vanishing Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>$\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)$</td>
<td>$n^{\frac{nr}{r+1} - k - 1}$</td>
<td></td>
</tr>
<tr>
<td>Field size</td>
<td>$n^{\frac{nr}{r+1} - k - 1}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **High-Rate Regime:** Gopalan, Huang, Jenkins, Yekhanin 14
- **Constant-Rate Regime:** Rawat, Koyluoglu, Silberstein, Vishwanath 14

Maximally Recoverable Codes
Summary of best known constructions of MR codes

<table>
<thead>
<tr>
<th></th>
<th>High-Rate</th>
<th>Constant-Rate</th>
<th>Vanishing Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimension</strong></td>
<td>( \frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right) )</td>
<td>( k ) ( \frac{n}{r+1} &lt; ) ( r )</td>
<td></td>
</tr>
<tr>
<td><strong>Field size</strong></td>
<td>( n^{\frac{nr}{r+1} - k - 1} )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **High-Rate Regime:** Gopalan, Huang, Jenkins, Yekhanin 14
- **Constant-Rate Regime:** Rawat, Koyluoglu, Silberstein, Vishwanath 14
Summary of best known constructions of MR codes

<table>
<thead>
<tr>
<th></th>
<th>High-Rate</th>
<th>Constant-Rate</th>
<th>Vanishing Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>$\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)$</td>
<td>$\frac{k}{n} &lt; \frac{r}{r+1}$</td>
<td></td>
</tr>
<tr>
<td>Field size</td>
<td>$n^{\frac{nr}{r+1} - k - 1}$</td>
<td>$2^{\frac{nr}{r+1}}$</td>
<td></td>
</tr>
</tbody>
</table>

- **High-Rate Regime:** Gopalan, Huang, Jenkins, Yekhanin 14
- **Constant-Rate Regime:** Rawat, Koyluoglu, Silberstein, Vishwanath 14
Summary of best known constructions of MR codes

<table>
<thead>
<tr>
<th></th>
<th>High-Rate</th>
<th>Constant-Rate</th>
<th>Vanishing Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimension</strong></td>
<td>$\frac{nr}{r+1} - k = O\left(\frac{n}{\log n}\right)$</td>
<td>$k/n &lt; \frac{r}{r+1}$</td>
<td></td>
</tr>
<tr>
<td><strong>Field size</strong></td>
<td>$n^{\frac{nr}{r+1} - k - 1}$</td>
<td>$2^{\frac{nr}{r+1}}$</td>
<td></td>
</tr>
</tbody>
</table>

- **High-Rate Regime:** Gopalan, Huang, Jenkins, Yekhanin 14
- **Constant-Rate Regime:** Rawat, Koyluoglu, Silberstein, Vishwanath 14
- **Vanishing Rate Regime:** T., Papailiopoulos, Dimakis 14
**Summary of best known constructions of MR codes**

<table>
<thead>
<tr>
<th>Dimension</th>
<th>High-Rate</th>
<th>Constant-Rate</th>
<th>Vanishing Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{nr}{r+1} - k = O(\frac{n}{\log n})$</td>
<td>$\frac{k}{n} &lt; \frac{r}{r+1}$</td>
<td>$k = O(\frac{n}{\log n})$</td>
<td></td>
</tr>
<tr>
<td>Field size</td>
<td>$n^{\frac{nr}{r+1} - k - 1}$</td>
<td>$2^{\frac{nr}{r+1}}$</td>
<td></td>
</tr>
</tbody>
</table>

- **High-Rate Regime:** Gopalan, Huang, Jenkins, Yekhanin 14
- **Constant-Rate Regime:** Rawat, Koyluoglu, Silberstein, Vishwanath 14
- **Vanishing Rate Regime:** T., Papailiopoulos, Dimakis 14
## Summary of best known constructions of MR codes

<table>
<thead>
<tr>
<th></th>
<th>High-Rate</th>
<th>Constant-Rate</th>
<th>Vanishing Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimension</strong></td>
<td>( \frac{nr}{r+1} - k = O\left( \frac{n}{\log n}\right) )</td>
<td>( \frac{k}{n} &lt; \frac{r}{r+1} )</td>
<td>( k = O\left( \frac{n}{\log n}\right) )</td>
</tr>
<tr>
<td><strong>Field size</strong></td>
<td>( n^{\frac{nr}{r+1} - k - 1} )</td>
<td>( 2^{\frac{nr}{r+1}} )</td>
<td>( n^k )</td>
</tr>
</tbody>
</table>

- **High-Rate Regime**: Gopalan, Huang, Jenkins, Yekhanin 14
- **Constant-Rate Regime**: Rawat, Koyluoglu, Silberstein, Vishwanath 14
- **Vanishing Rate Regime**: T., Papailiopoulos, Dimakis 14
Lower bounds on the field size

\[ F(n, k, r) \text{ is upper bounded by a superpolynomial function of } n \]

**Theorem [Gopalan, Huang, Jenkins, Yekhanin 14]**

\[ F(n, k, r) \geq k + 1 \]

**Proof:**
- Puncturing the code by one coordinate from each recovery set results in an \((nr + 1, k)\) MDS code.
- By the bound on the field size of an MDS code:
  \[ nr + 1 \leq |F| + (nr + 1 - k) - 1 \]

**Questions:**
1. What is the smallest field size needed for an \((n, k, r)\) MR code?
   - For optimal LRC order, \(n\) is sufficient.
   - Contains the MDS conjecture as a subproblem.
2. Are MR codes inherently different from optimal LRC codes?
Lower bounds on the field size

$F(n, k, r)$ is upper bounded by a superpolynomial function of $n$

**Theorem** [Gopalan, Huang, Jenkins, Yekhanin 14]

**Proof:**

- Puncturing the code by one coordinate from each recovery set results in an $(nr + 1, k)$ MDS code
- By the bound on the field size of an MDS code, $nr + 1 \leq |F| + (nr + 1 - k) - 1$

**Questions:**

1. What is the smallest field size needed for an $(n, k, r)$ MR code?
   - For optimal LRC order, $n$ is sufficient
   - Contains the MDS conjecture as a subproblem

2. Are MR codes inherently different from optimal LRC codes?
Lower bounds on the field size

\( F(n, k, r) \) is upper bounded by a superpolynomial function of \( n \)

**Theorem** [Gopalan, Huang, Jenkins, Yekhanin 14]

\[ F(n, k, r) \geq k + 1 \]
Lower bounds on the field size

\( F(n, k, r) \) is upper bounded by a superpolynomial function of \( n \)

**Theorem** [Gopalan, Huang, Jenkins, Yekhanin 14]

\[ F(n, k, r) \geq k + 1 \]

**Proof:**
Lower bounds on the field size

\( F(n, k, r) \) is upper bounded by a superpolynomial function of \( n \)

**Theorem** [Gopalan, Huang, Jenkins, Yekhanin 14]

\[ F(n, k, r) \geq k + 1 \]

**Proof:**

- Puncturing the code by one coordinate from each recovery set results an \((\frac{nr}{r+1}, k)\) MDS code
Lower bounds on the field size

\( \mathbb{F}(n, k, r) \) is upper bounded by a superpolynomial function of \( n \)

**Theorem** [Gopalan, Huang, Jenkins, Yekhanin 14]

\[ \mathbb{F}(n, k, r) \geq k + 1 \]

**Proof:**

- Puncturing the code by one coordinate from each recovery set results an \( \left( \frac{nr}{r+1}, k \right) \) MDS code
- By the bound on the field size of an MDS code \( \frac{nr}{r+1} \leq |\mathbb{F}| + \left( \frac{nr}{r+1} - k \right) - 1 \)
Lower bounds on the field size

$\mathbb{F}(n, k, r)$ is upper bounded by a superpolynomial function of $n$

**Theorem** [Gopalan, Huang, Jenkins, Yekhanin 14]

$\mathbb{F}(n, k, r) \geq k + 1$

**Proof:**

- Puncturing the code by one coordinate from each recovery set results an $(\frac{nr}{r+1}, k)$ MDS code
- By the bound on the field size of an MDS code $\frac{nr}{r+1} \leq |\mathbb{F}| + (\frac{nr}{r+1} - k) - 1$

**Questions:**

1. What is the smallest field size needed for an $(n, k, r)$ MR code?
   - For optimal LRC order $n$ is sufficient
   - Contains the MDS conjecture as a subproblem
2. Are MR codes inherently different from optimal LRC codes?
Lower bounds on the field size

\( \mathbb{F}(n, k, r) \) is upper bounded by a superpolynomial function of \( n \)

**Theorem** [Gopalan, Huang, Jenkins, Yekhanin 14]

\[ \mathbb{F}(n, k, r) \geq k + 1 \]

**Proof:**

- Puncturing the code by one coordinate from each recovery set results an \( \left( \frac{nr}{r+1}, k \right) \) MDS code
- By the bound on the field size of an MDS code
  \[ \frac{nr}{r+1} \leq |\mathbb{F}| + \left( \frac{nr}{r+1} - k \right) - 1 \]

**Questions:**

1. What is the smallest field size needed for an \( (n, k, r) \) MR code?
Lower bounds on the field size

\( \mathbb{F}(n, k, r) \) is upper bounded by a superpolynomial function of \( n \)

**Theorem** [Gopalan, Huang, Jenkins, Yekhanin 14]

\[ \mathbb{F}(n, k, r) \geq k + 1 \]

**Proof:**

- Puncturing the code by one coordinate from each recovery set results an \( \left( \frac{nr}{r+1}, k \right) \) MDS code
- By the bound on the field size of an MDS code \( \frac{nr}{r+1} \leq |\mathbb{F}| + \left( \frac{nr}{r+1} - k \right) - 1 \)

**Questions:**

1. What is the smallest field size needed for an \( (n, k, r) \) MR code?
   - For optimal LRC order \( n \) is sufficient
Lower bounds on the field size

\( \mathbb{F}(n, k, r) \) is upper bounded by a superpolynomial function of \( n \)

**Theorem** [Gopalan, Huang, Jenkins, Yekhanin 14]

\[
\mathbb{F}(n, k, r) \geq k + 1
\]

**Proof:**

- Puncturing the code by one coordinate from each recovery set results an \((\frac{nr}{r+1}, k)\) MDS code
- By the bound on the field size of an MDS code \( \frac{nr}{r+1} \leq |\mathbb{F}| + (\frac{nr}{r+1} - k) - 1 \)

**Questions:**

1. What is the smallest field size needed for an \((n, k, r)\) MR code?
   - For optimal LRC order \( n \) is sufficient
   - Contains the MDS conjecture as a subproblem
Lower bounds on the field size

\( \mathbb{F}(n, k, r) \) is upper bounded by a superpolynomial function of \( n \)

**Theorem** [Gopalan, Huang, Jenkins, Yekhanin 14]

\[
\mathbb{F}(n, k, r) \geq k + 1
\]

**Proof:**

- Puncturing the code by one coordinate from each recovery set results an \((\frac{nr}{r+1}, k)\) MDS code
- By the bound on the field size of an MDS code \( \frac{nr}{r+1} \leq |\mathbb{F}| + \left( \frac{nr}{r+1} - k \right) - 1 \)

**Questions:**

1. What is the smallest field size needed for an \((n, k, r)\) MR code?
   - For optimal LRC order \( n \) is sufficient
   - Contains the MDS conjecture as a subproblem
2. Are MR codes inherently different from optimal LRC codes?
MR codes under arbitrary topologies

• $R = \{ R_i \subseteq [n] : i = 1, \ldots, m \}$ constraints on code's coordinates,

• $C_R$ is a code with a set of coordinate constraints $R$.

• Definition: Given a code $C_R$, its Information sets $I = \{ S \subseteq [n] : C_R \text{ is decodable from } S, |S| = k \}$.

• Goal: Construct a code $C_R$ with local constraints $R$ that is "highly" decodable (large set $I$).

• Definition: $C_R$ with information sets $I$ is MR if for any $C'_R$ with $I' \subseteq I$ (i.e. $I$ is a maximum).

Maximally Recoverable Codes 15 / 17
MR codes under arbitrary topologies

- \( \mathcal{R} = \{ \mathcal{R}_i \subseteq [n] : i = 1, \ldots, m \} \) constraints on code’s coordinates,
MR codes under arbitrary topologies

- $\mathcal{R} = \{\mathcal{R}_i \subseteq [n] : i = 1, \ldots, m\}$ constraints on code’s coordinates,

1. E.g. $\mathcal{R}_1 = \{1, \ldots, r + 1\}$, $\mathcal{R}_2 = \{r + 2, \ldots, 2(r + 1)\}$, \ldots, $\mathcal{R}_{\frac{nr}{r+1}}$
MR codes under arbitrary topologies

• $\mathcal{R} = \{ \mathcal{R}_i \subseteq [n] : i = 1, \ldots, m \}$ constraints on code’s coordinates,

1. E.g. $\mathcal{R}_1 = \{1, \ldots, r + 1\}$, $\mathcal{R}_2 = \{r + 2, \ldots, 2(r + 1)\}$, ..., $\mathcal{R}_{\frac{nr}{r+1}}$

2. Can be arbitrary code constraints
MR codes under arbitrary topologies

- $\mathcal{R} = \{ \mathcal{R}_i \subseteq [n] : i = 1, ..., m \}$ constraints on code’s coordinates,

- $C_\mathcal{R}$ = a code with a set of coordinate constraints $\mathcal{R}$
MR codes under arbitrary topologies

- $\mathcal{R} = \{ \mathcal{R}_i \subseteq [n] : i = 1, \ldots, m \}$ constraints on code’s coordinates,

- $C_{\mathcal{R}}$ = a code with a set of coordinate constraints $\mathcal{R}$

- **Definition:** Given a code $C_{\mathcal{R}}$, its Information sets

  $$\mathcal{I} = \{ S \subseteq [n] : C_{\mathcal{R}} \text{ is decodable from } S, |S| = k \}$$
MR codes under arbitrary topologies

- $\mathcal{R} = \{\mathcal{R}_i \subseteq [n] : i = 1, \ldots, m\}$ constraints on code’s coordinates,

- $\mathcal{C}_\mathcal{R}$ = a code with a set of coordinate constraints $\mathcal{R}$

- **Definition:** Given a code $\mathcal{C}_\mathcal{R}$, its Information sets

  \[ I = \{S \subseteq [n] : \mathcal{C}_\mathcal{R} \text{ is decodable from } S, |S| = k\} \]

- **Goal:** Construct a code $\mathcal{C}_\mathcal{R}$ with local constraints $\mathcal{R}$ that is “highly” decodable (large set $I$)
MR codes under arbitrary topologies

- $\mathcal{R} = \{\mathcal{R}_i \subseteq [n] : i = 1, \ldots, m\}$ constraints on code’s coordinates,

- $C_\mathcal{R} =$ a code with a set of coordinate constraints $\mathcal{R}$

**Definition:** Given a code $C_\mathcal{R}$, its Information sets

$$\mathcal{I} = \{S \subseteq [n] : C_\mathcal{R} \text{ is decodable from } S, |S| = k\}$$

**Goal:** Construct a code $C_\mathcal{R}$ with local constraints $\mathcal{R}$ that is “highly” decodable
(large set $\mathcal{I}$)

**Definition:** $C_\mathcal{R}$ with information sets $\mathcal{I}$ is MR if for any $C'_\mathcal{R}$ with $\mathcal{I}'$

$$\mathcal{I}' \subseteq \mathcal{I} \text{ (i.e. } \mathcal{I} \text{ is a maximum)}$$
MR codes under arbitrary topologies

- **Definition:** $C_{\mathcal{R}}$ with $\mathcal{I}$ is MR if for any $C'_{\mathcal{R}}$ with $\mathcal{I}'$

  \[
  \mathcal{I}' \subseteq \mathcal{I}
  \]
MR codes under arbitrary topologies

- **Definition:** \( C_R \) with \( I \) is MR if for any \( C'_R \) with \( I' \)

\[
I' \subseteq I
\]

- **Example:**
MR codes under arbitrary topologies

- **Definition:** $C_{\mathcal{R}}$ with $\mathcal{I}$ is MR if for any $C_{\mathcal{R}}'$ with $\mathcal{I}'$
  
  $\mathcal{I}' \subseteq \mathcal{I}$

- **Example:** $\mathcal{R}_1 = \{1, \ldots, r + 1\}$
MR codes under arbitrary topologies

- **Definition:** $C_R$ with $I$ is MR if for any $C'_R$ with $I'$
  \[ I' \subseteq I \]

- **Example:** $\mathcal{R}_1 = \{1, \ldots, r + 1\}$, $\ldots$, $\mathcal{R}_{n/(r+1)} = \{n - r, \ldots, n\}$
MR codes under arbitrary topologies

- **Definition:** $C_R$ with $I$ is MR if for any $C'_R$ with $I'$

  $$I' \subseteq I$$

- **Example:** $R_1 = \{1, \ldots, r + 1\}, \ldots, R_{n/(r+1)} = \{n - r, \ldots, n\}$

  $C_R$ is MR if $I = \{S \subseteq [n] : R_i \not\subseteq S, |S| = k\}$
MR codes under arbitrary topologies

- **Definition:** $C_{\mathcal{R}}$ with $\mathcal{I}$ is MR if for any $C'_{\mathcal{R}}$ with $\mathcal{I}'$
  \[
  \mathcal{I}' \subseteq \mathcal{I}
  \]

- **Example:** $\mathcal{R}_1 = \{1, \ldots, r + 1\}$, ..., $\mathcal{R}_{n/(r+1)} = \{n - r, \ldots, n\}$
  
  $C_{\mathcal{R}}$ is MR if $\mathcal{I} = \{S \subseteq [n] : \mathcal{R}_i \not\subseteq S, |S| = k\}$

**Lemma**

For any set $\mathcal{R}$ there exists an MR code $C_{\mathcal{R}}$ over a large enough field
MR codes under arbitrary topologies - cont’d

“Maximally Recoverable Codes for Grid-like Topologies” [Gopalan, Hu, Saraf, Wang, Yekhanin 16]
MR codes under arbitrary topologies - cont’d

“Maximally Recoverable Codes for Grid-like Topologies” [Gopalan, Hu, Saraf, Wang, Yekhanin 16]

1. Good problem introduction
“Maximally Recoverable Codes for Grid-like Topologies” [Gopalan, Hu, Saraf, Wang, Yekhanin 16]

1. Good problem introduction

2. The \textit{first} non-trivial lower bound on $|F|$ (for grid-like topologies)
MR codes under arbitrary topologies - cont’d

“Maximally Recoverable Codes for Grid-like Topologies” [Gopalan, Hu, Saraf, Wang, Yekhanin 16]

1. Good problem introduction

2. The first non-trivial lower bound on $|F|$ (for grid-like topologies)

3. More results...
“Maximally Recoverable Codes for Grid-like Topologies” [Gopalan, Hu, Saraf, Wang, Yekhanin 16]

1. Good problem introduction

2. The *first* non-trivial lower bound on $|F|$ (for grid-like topologies)

3. More results...

Questions:
MR codes under arbitrary topologies - cont’d

“Maximally Recoverable Codes for Grid-like Topologies” [Gopalan, Hu, Saraf, Wang, Yekhanin 16]

1. Good problem introduction

2. The *first* non-trivial lower bound on $|\mathbb{F}|$ (for grid-like topologies)

3. More results...

Questions:

1. Given an MR code with set of constraints $\mathcal{R}$ characterize its information sets $\mathcal{I}$
MR codes under arbitrary topologies - cont’d

“Maximally Recoverable Codes for Grid-like Topologies” [Gopalan, Hu, Saraf, Wang, Yekhanin 16]

1. Good problem introduction

2. The *first* non-trivial lower bound on $|F|$ (for grid-like topologies)

3. More results...

Questions:

1. Given an MR code with set of constraints $\mathcal{R}$ characterize its information sets $\mathcal{I}$

2. MR code constructions for other topologies (e.g. grid-like topologies)
MR codes under arbitrary topologies - cont’d

“Maximally Recoverable Codes for Grid-like Topologies” [Gopalan, Hu, Saraf, Wang, Yekhanin 16]

1. Good problem introduction

2. The first non-trivial lower bound on $|\mathcal{F}|$ (for grid-like topologies)

3. More results...

Questions:

1. Given an MR code with set of constraints $\mathcal{R}$ characterize its information sets $\mathcal{I}$

2. MR code constructions for other topologies (e.g. grid-like topologies)

3. Are the known constructions extendable to other topologies?
MR codes under arbitrary topologies - cont’d

“Maximally Recoverable Codes for Grid-like Topologies” [Gopalan, Hu, Saraf, Wang, Yekhanin 16]

1. Good problem introduction

2. The *first* non-trivial lower bound on $|\mathbb{F}|$ (for grid-like topologies)

3. More results...

Questions:

1. Given an MR code with set of constraints $\mathcal{R}$ characterize its information sets $\mathcal{I}$

2. MR code constructions for other topologies (e.g. grid-like topologies)

3. Are the known constructions extendable to other topologies?

4. Field size?
MR codes under arbitrary topologies - cont’d

“Maximally Recoverable Codes for Grid-like Topologies” [Gopalan, Hu, Saraf, Wang, Yekhanin 16]

1. Good problem introduction

2. The first non-trivial lower bound on $|\mathbb{F}|$ (for grid-like topologies)

3. More results...

Questions:

1. Given an MR code with set of constraints $\mathcal{R}$ characterize its information sets $\mathcal{I}$

2. MR code constructions for other topologies (e.g. grid-like topologies)

3. Are the known constructions extendable to other topologies?

4. Field size?

5. Is it possible to construct MR codes using the RS-like construction?
LRC codes on graphs

In this part we used materials provided to us by Fatemeh Arbajolfaei (University of California, San Diego), Søren Riis (Queen Mary, University of London) and Arya Mazumdar (University of Massachusetts at Amherst), with their kind permission.
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

$x_1$

$x_2$

$x_3$

$G$
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]

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

![Storage recovery graph $G$](image.png)
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:
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 $C \subseteq \mathbb{F}_q^n$
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 $C \subseteq \mathbb{F}_q^n$
  2. $n$ recovery functions $f_i$, s.t. for any $(x_1, ..., x_n) \in C$
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 \( C \subseteq \mathbb{F}_q^n \)
  2. \( n \) recovery functions \( f_i \), s.t. for any \( (x_1, \ldots, x_n) \in C \)
     \[ f_i(x_j : j \in N(i)) = x_i \]
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 $C \subseteq \mathbb{F}_q^n$
  2. $n$ recovery functions $f_i$, s.t. for any $(x_1, ..., x_n) \in C$
     \[ f_i(x_j : j \in N(i)) = x_i \]
- The storage capacity of $G$ over $\mathbb{F}_q$
  \[ \text{Cap}_q(G) = \max_{C \subseteq \mathbb{F}_q^n \text{ is a storage code for } G} \log_q |C| \]
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 $C \subseteq \mathbb{F}_q^n$
  2. $n$ recovery functions $f_i$, s.t. for any $(x_1, \ldots, x_n) \in C$
     $$f_i(x_j : j \in N(i)) = x_i$$

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

$$\text{Cap}_q(G) = \max_{C \subseteq \mathbb{F}_q^n \text{ is a storage code for } G} \log_q |C| \leq n$$
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 $C \subseteq \mathbb{F}_q^n$
  2. $n$ recovery functions $f_i$, s.t. for any $(x_1, \ldots, x_n) \in C$
     $$f_i(x_j : j \in N(i)) = x_i$$

- The storage capacity of $G$ over $\mathbb{F}_q$
  $$\text{Cap}_q(G) = \max_{\text{C is a storage code for } G} \log_q |C| \leq n$$

- The storage capacity of $G$ is
  $$\text{Cap}(G) = \sup_q \text{Cap}_q(G)$$
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 $C \subseteq \mathbb{F}_q^n$
  2. $n$ recovery functions $f_i$, s.t. for any $(x_1, \ldots, x_n) \in C$

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

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

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

- The storage capacity of $G$ is

$$\text{Cap}(G) = \sup_q \text{Cap}_q(G) = \lim_{q \to \infty} \text{Cap}_q(G) \text{ (Fekete’s lemma)}$$
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 ($\text{Cap}_2(G) = 2$)
- Store:
  - $N_1 = b_1$
  - $N_2 = b_2$
  - $N_3 = b_1 + b_2$
LRC codes on graphs - Example

- Each node stores a single bit
- Two bits of information $b_1, b_2$ can be stored ($\text{Cap}_2(G) = 2$)
- Store:
  - $N_1 = b_1$
  - $N_2 = b_2$
  - $N_3 = b_1 + b_2$
- $\text{Cap}(G) = \sup_q \text{Cap}_q(G)$
LRC codes on graphs - Example

- Each node stores a single bit
- Two bits of information $b_1, b_2$ can be stored ($\text{Cap}_2(G) = 2$)
- Store:
  - $N_1 = b_1$
  - $N_2 = b_2$
  - $N_3 = b_1 + b_2$
- $\text{Cap}(G) = \sup_q \text{Cap}_q(G) = 2$
LRC codes on graphs - Example
LRC codes on graphs - Example

- $C = \{(0, 0, 0, 0, 0), (0, 1, 1, 0, 0), (0, 0, 0, 1, 1), (1, 1, 0, 1, 1), (1, 1, 1, 0, 1)\}$
LRC codes on graphs - Example

LRC codes

\[ C = \{(0,0,0,0,0), (0,1,1,0,0), (0,0,0,1,1), (1,1,0,1,1), (1,1,1,0,1)\} \]

\[ N_1 = N_2 \land N_5, \]
LRC codes on graphs - Example

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

- $N_1 = N_2 \land N_5, N_2 = N_1 \lor N_3, N_3 = N_2 \land \overline{N_4}, N_4 = \overline{N_3} \land N_5, N_5 = N_1 \lor N_4$
LRC codes on graphs - Example

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

- $N_1 = N_2 \land N_5, N_2 = N_1 \lor N_3, N_3 = N_2 \land \overline{N_4}, N_4 = \overline{N_3} \land N_5, N_5 = N_1 \lor N_4$

- $\text{Cap}_2(G) \geq \log_2(5) = 2.32...$
LRC codes on graphs - Example

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

- $N_1 = N_2 \land N_5, N_2 = N_1 \lor N_3, N_3 = N_2 \land \overline{N_4}, N_4 = \overline{N_3} \land N_5, N_5 = N_1 \lor N_4$

- $\text{Cap}_2(G) \geq \log_2(5) = 2.32\ldots$

- In fact $\text{Cap}_2(G) = \log_2(5) = 2.32\ldots$
\[ C = \{(0, 0, 0, 0, 0), (0, 1, 1, 0, 0), (0, 0, 0, 1, 1), (1, 1, 0, 1, 1), (1, 1, 1, 0, 1)\} \]

\[ N_1 = N_2 \land N_5, N_2 = N_1 \lor N_3, N_3 = N_2 \land \overline{N}_4, N_4 = \overline{N}_3 \land N_5, N_5 = N_1 \lor N_4 \]

\[ \text{Cap}_2(G) \geq \log_2(5) = 2.32... \]

\[ \text{In fact} \quad \text{Cap}_2(G) = \log_2(5) = 2.32... \]

\[ \text{However} \quad \text{Cap}(G) = \text{Cap}_4(G) \]
LRC codes on graphs - Example

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

- \( N_1 = N_2 \land N_5, N_2 = N_1 \lor N_3, N_3 = N_2 \land \overline{N_4}, N_4 = \overline{N_3} \land N_5, N_5 = N_1 \lor N_4 \)

- \( \text{Cap}_2(G) \geq \log_2(5) = 2.32... \)

- 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]
LRC codes on graphs - Example
LRC codes on graphs - Example

- $\text{Cap}_4(G) = 2.5 \rightarrow 5$ bits can be stored in the system $\{1, 2, 3, 4, 5\}$
LRC codes on graphs - Example

- $\text{Cap}_4(G) = 2.5 \rightarrow 5$ bits can be stored in the system $\{1, 2, 3, 4, 5\}$
LRC codes on graphs - Example

- $\text{Cap}_4(G) = 2.5 \rightarrow 5$ bits can be stored in the system \{1, 2, 3, 4, 5\}
LRC codes on graphs - Example

- $\text{Cap}_4(G) = 2.5 \rightarrow 5$ bits can be stored in the system $\{1, 2, 3, 4, 5\}$
LRC codes on graphs - Example

- \( \text{Cap}_4(G) = 2.5 \rightarrow 5 \) bits can be stored in the system \{1, 2, 3, 4, 5\}
LRC codes on graphs - Example

- $\text{Cap}_4(G) = 2.5 \rightarrow 5$ bits can be stored in the system $\{1, 2, 3, 4, 5\}$
LRC codes on graphs - Example

- $\text{Cap}_4(G) = 2.5 \rightarrow 5$ bits can be stored in the system \{1, 2, 3, 4, 5\}

- $\text{Cap}(G) \leq 2.5$?
LRC codes on graphs - Example

- $H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z)$
LRC codes on graphs - Example

• $H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z)$

$H(N_1, N_2, N_3, N_4, N_5)$
LRC codes on graphs - Example

- \( H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z) \)

\[
H(N_1, N_2, N_3, N_4, N_5) = H(N_1, N_2, N_3, N_4, N_5) + H(N_4, N_5) - H(N_4, N_5)
\]
LRC codes on graphs - Example

- \( H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z) \)

\[
H(N_1, N_2, N_3, N_4, N_5) = H(N_1, N_2, N_3, N_4, N_5) + H(N_4, N_5) - H(N_4, N_5)
\]

\[
= H(N_1, N_3, N_4, N_5) + H(N_4, N_5) - H(N_4, N_5)
\]
LRC codes on graphs - Example

- $H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z)$

\[
H(N_1, N_2, N_3, N_4, N_5) = H(N_1, N_2, N_3, N_4, N_5) + H(N_4, N_5) - H(N_4, N_5) \\
= H(N_1, N_3, N_4, N_5) + H(N_4, N_5) - H(N_4, N_5) \\
\leq H(N_1, N_4, N_5) + H(N_3, N_4, N_5) - H(N_4, N_5)
\]
LRC codes on graphs - Example

• \( H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z) \)

\[
H(N_1, N_2, N_3, N_4, N_5) = H(N_1, N_2, N_3, N_4, N_5) + H(N_4, N_5) - H(N_4, N_5) \\
= H(N_1, N_3, N_4, N_5) + H(N_4, N_5) - H(N_4, N_5) \\
\leq H(N_1, N_4, N_5) + H(N_3, N_4, N_5) - H(N_4, N_5) \\
= H(N_1, N_4) + H(N_3, N_5) - H(N_4, N_5)
\]
LRC codes on graphs - Example

- \( H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z) \)

\[
H(N_1, N_2, N_3, N_4, N_5) = H(N_1, N_2, N_3, N_4, N_5) + H(N_4, N_5) - H(N_4, N_5)
\]

\[
= H(N_1, N_3, N_4, N_5) + H(N_4, N_5) - H(N_4, N_5)
\]

\[
\leq H(N_1, N_4, N_5) + H(N_3, N_4, N_5) - H(N_4, N_5)
\]

\[
= H(N_1, N_4) + H(N_3, N_5) - H(N_4, N_5)
\]

\[
\leq H(N_1) + H(N_3) + H(N_4) + H(N_5) - H(N_4, N_5)
\]
LRC codes on graphs - Example

- \( H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z) \)

- \( H(N_1, N_2, N_3, N_4, N_5) \leq H(N_1) + H(N_3) + H(N_4) + H(N_5) - H(N_4, N_5) \)
LRC codes on graphs - Example

- \( H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z) \)

- \( H(N_1, N_2, N_3, N_4, N_5) \leq H(N_1) + H(N_3) + H(N_4) + H(N_5) - H(N_4, N_5) \)

\[
H(N_1, N_2, N_3, N_4, N_5) = H(N_2, N_4, N_5)
\]
LRC codes on graphs - Example

- $H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z)$

- $H(N_1, N_2, N_3, N_4, N_5) \leq H(N_1) + H(N_3) + H(N_4) + H(N_5) - H(N_4, N_5)$

\[
\begin{align*}
H(N_1, N_2, N_3, N_4, N_5) &= H(N_2, N_4, N_5) \\
&= H(N_4, N_5) + H(N_2 | N_4, N_5)
\end{align*}
\]
LRC codes on graphs - Example

- \( H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z) \)

- \( H(N_1, N_2, N_3, N_4, N_5) \leq H(N_1) + H(N_3) + H(N_4) + H(N_5) - H(N_4, N_5) \)

\[
H(N_1, N_2, N_3, N_4, N_5) = H(N_2, N_4, N_5) \\
= H(N_4, N_5) + H(N_2|N_4, N_5) \\
\leq H(N_4, N_5) + H(N_2)
\]
LRC codes on graphs - Example

- $H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z)$

- $H(N_1, N_2, N_3, N_4, N_5) \leq H(N_1) + H(N_3) + H(N_4) + H(N_5) - H(N_4, N_5)$

- $H(N_1, N_2, N_3, N_4, N_5) \leq H(N_4, N_5) + H(N_2)$
LRC codes on graphs - Example

- \( H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z) \)

- \( H(N_1, N_2, N_3, N_4, N_5) \leq H(N_1) + H(N_3) + H(N_4) + H(N_5) - H(N_4, N_5) \)

- \( H(N_1, N_2, N_3, N_4, N_5) \leq H(N_4, N_5) + H(N_2) \)

- \( \Rightarrow 2H(N_1, N_2, N_3, N_4, N_5) \leq H(N_1) + H(N_2) + H(N_3) + H(N_4) + H(N_5) \leq 5 \)
LRC codes on graphs - Example

- \( H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z) \)

- \( H(N_1, N_2, N_3, N_4, N_5) \leq H(N_1) + H(N_3) + H(N_4) + H(N_5) - H(N_4, N_5) \)

- \( H(N_1, N_2, N_3, N_4, N_5) \leq H(N_4, N_5) + H(N_2) \)

- \( \Rightarrow 2H(N_1, N_2, N_3, N_4, N_5) \leq H(N_1) + H(N_2) + H(N_3) + H(N_4) + H(N_5) \leq 5 \)

- \( \Rightarrow \text{Cap}(G) \leq 2.5 \)
LRC codes on graphs - Example

- \( H(X, Y, Z) + H(Z) \leq H(X, Z) + H(Y, Z) \)

- \( H(N_1, N_2, N_3, N_4, N_5) \leq H(N_1) + H(N_3) + H(N_4) + H(N_5) - H(N_4, N_5) \)

- \( H(N_1, N_2, N_3, N_4, N_5) \leq H(N_4, N_5) + H(N_2) \)

- \( \Rightarrow 2H(N_1, N_2, N_3, N_4, N_5) \leq H(N_1) + H(N_2) + H(N_3) + H(N_4) + H(N_5) \leq 5 \)

- \( \Rightarrow \text{Cap}(G) \leq 2.5 \)

- **Question:** Given an arbitrary graph \( G \) how to calculate \( \text{Cap}(G) \)?
The Confusion Graph [Bar-Yossef, Birk, Jayram, and Kol 2006]

Confusion Graph
Confusion Graph

- Given a graph $G$ on $n$ vertices and $q \in \mathbb{N}$
The Confusion Graph [Bar-Yossef, Birk, Jayram, and Kol 2006]

Confusion Graph

- Given a graph $G$ on $n$ vertices and $q \in \mathbb{N}$
- The confusion graph $\text{Conf}(G)$ has
The Confusion Graph [Bar-Yossef, Birk, Jayram, and Kol 2006]

Confusion Graph

• Given a graph $G$ on $n$ vertices and $q \in \mathbb{N}$

• The confusion graph $\text{Conf}(G)$ has
  
  • **Vertex set:** $\mathbb{F}_q^n$
The Confusion Graph [Bar-Yossef, Birk, Jayram, and Kol 2006]

Confusion Graph

- Given a graph $G$ on $n$ vertices and $q \in \mathbb{N}$
- The confusion graph $\text{Conf}(G)$ has
  - **Vertex set**: $\mathbb{F}_q^n$
  - **Edge set**: $v = (v_1, \ldots, v_n) \sim u = (u_1, \ldots, u_n)$ if
The Confusion Graph [Bar-Yossef, Birk, Jayram, and Kol 2006]

Confusion Graph

- Given a graph $G$ on $n$ vertices and $q \in \mathbb{N}$
- The confusion graph $\text{Conf}(G)$ has

  - **Vertex set:** $\mathbb{F}_q^n$
  - **Edge set:** $v = (v_1, \ldots, v_n) \sim u = (u_1, \ldots, u_n)$ if
    1. Exists $i$ s.t. $v_i \neq u_i$ and
The Confusion Graph [Bar-Yossef, Birk, Jayram, and Kol 2006]

Confusion Graph

- Given a graph $G$ on $n$ vertices and $q \in \mathbb{N}$
- The confusion graph $\text{Conf}(G)$ has
  - **Vertex set:** $\mathbb{F}_q^n$
  - **Edge set:** $v = (v_1, ..., v_n) \sim u = (u_1, ..., u_n)$ if
    1. Exists $i$ s.t. $v_i \neq u_i$ and
    2. $v_{N(i)} = u_{N(i)}$
The Confusion Graph [Bar-Yossef, Birk, Jayram, and Kol 2006]

Confusion Graph

- Given a graph $G$ on $n$ vertices and $q \in \mathbb{N}$
- The confusion graph $\text{Conf}(G)$ has
  
  - **Vertex set:** $\mathbb{F}_q^n$
  
  - **Edge set:** $v = (v_1, ..., v_n) \sim u = (u_1, ..., u_n)$ if
    
    1. Exists $i$ s.t. $v_i \neq u_i$ and
    2. $v|_{N(i)} = u|_{N(i)}$

- $v, u \in \mathbb{F}_q^n$ can **both** be contained in a storage code $\Leftrightarrow v \sim u$ in $\text{Conf}(G)$
The Confusion Graph [Bar-Yossef, Birk, Jayram, and Kol 2006]

Confusion Graph

- Given a graph $G$ on $n$ vertices and $q \in \mathbb{N}$
- The confusion graph $\text{Conf}(G)$ has
  - **Vertex set:** $\mathbb{F}_q^n$
  - **Edge set:** $v = (v_1, \ldots, v_n) \sim u = (u_1, \ldots, u_n)$ if
    1. Exists $i$ s.t. $v_i \neq u_i$ and
    2. $v|_{\mathcal{N}(i)} = u|_{\mathcal{N}(i)}$

- $v, u \in \mathbb{F}_q^n$ can both be contained in a storage code $\iff v \sim u$ in $\text{Conf}(G)$
- $C \subseteq \mathbb{F}_q^n$ is a storage code $\iff C$ is an independent set of $\text{Conf}(G)$
Confusion Graph

- Given a graph $G$ on $n$ vertices and $q \in \mathbb{N}$
- The confusion graph $\text{Conf}(G)$ has
  - **Vertex set:** $\mathbb{F}_q^n$
  - **Edge set:** $v = (v_1, \ldots, v_n) \sim u = (u_1, \ldots, u_n)$ if
    1. Exists $i$ s.t. $v_i \neq u_i$ and
    2. $v|_{N(i)} = u|_{N(i)}$

- $v, u \in \mathbb{F}_q^n$ can both be contained in a storage code $\iff v \sim u$ in $\text{Conf}(G)$
- $C \subseteq \mathbb{F}_q^n$ is a storage code $\iff C$ is an independent set of $\text{Conf}(G)$
- Corollary:
  $$\text{Cap}_q(G) = \log_q \alpha(\text{Conf}(G)),$$
  where $\alpha(\cdot)$ is the independence number
Computing $\text{Cap}(G)$

- $\text{Cap}_q(G) = \log_q \alpha(\text{Conf}(G))$
Computing $\text{Cap}(G)$

- $\text{Cap}_q(G) = \log_q \alpha(\text{Conf}(G))$

- Computing the independence number is hard
Computing $\text{Cap}(G)$

- $\text{Cap}_q(G) = \log_q \alpha(\text{Conf}(G))$
- Computing the independence number is hard
- Need to compute it for arbitrarily large confusion graphs ($q \to \infty$)
Computing $\text{Cap}(G)$

- $\text{Cap}_q(G) = \log_q \alpha(\text{Conf}(G))$
- Computing the independence number is hard
- Need to compute it for arbitrarily large confusion graphs ($q \rightarrow \infty$)
- Bounds on $\text{Cap}(G)$?
Index Coding [Birk, Kol 1998]
Index Coding [Birk, Kol 1998]

Side information $A_1 = \{2\}, A_2 = \{1, 3\}, A_3 = \{1\}$
Index Coding [Birk, Kol 1998]

- Side information graph $G$

- Side information $A_1 = \{2\}$, $A_2 = \{1, 3\}$, $A_3 = \{1\}$
Index Coding [Birk, Kol 1998]

- Side information graph $G$

- Side information $A_1 = \{2\}$, $A_2 = \{1, 3\}$, $A_3 = \{1\}$

- Send $x_1 + x_2$ and $x_3$
Index Coding [Birk, Kol 1998]

- Side information graph $G$

- Side information $A_1 = \{2\}$, $A_2 = \{1, 3\}$, $A_3 = \{1\}$

- Send $x_1 + x_2$ and $x_3$

- What is the fundamental limit on the number of transmissions?
Index Coding [Birk, Kol 1998]

- Side information graph $G$

- Side information $A_1 = \{2\}$, $A_2 = \{1, 3\}$, $A_3 = \{1\}$

- Send $x_1 + x_2$ and $x_3$

- What is the fundamental limit on the number of transmissions?

- Which coding scheme achieves the limit?

LRC codes
Index Coding [Birk, Kol 1998]

- Side information graph $G$

- Side information $A_1 = \{2\}$, $A_2 = \{1, 3\}$, $A_3 = \{1\}$

- Send $x_1 + x_2$ and $x_3$

- What is the fundamental limit on the number of transmissions?

- Which coding scheme achieves the limit?

- Many generalizations...
Index Coding [Birk, Kol 1998]

- $G = (V, E), |V| = n$ receivers

- Side information graph $G$

$LRC$ codes
Index Coding [Birk, Kol 1998]

- Side information graph $G$

- $G = (V, E), |V| = n$ receivers

- Receiver $i$ requests message $x_i \in \mathbb{F}_q$
Index Coding [Birk, Kol 1998]

- Side information graph $G$

- $G = (V, E), |V| = n$ receivers
- Receiver $i$ requests message $x_i \in F_q$
- An incoming edge represents the side information
Index Coding [Birk, Kol 1998]

- Side information graph $G$

- $G = (V, E)$, $|V| = n$ receivers

- Receiver $i$ requests message $x_i \in \mathbb{F}_q$

- An incoming edge represents the side information

- An index code of length $L$ is

LRC codes
Index Coding [Birk, Kol 1998]

- $G = (V, E), |V| = n$ receivers
- Receiver $i$ requests message $x_i \in \mathbb{F}_q$
- An incoming edge represents the side information
- An index code of length $L$ is
  1. An encoding function $E : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^L$

Side information graph $G$
**Index Coding** [Birk, Kol 1998]

- \( G = (V, E), |V| = n \) receivers
- Receiver \( i \) requests message \( x_i \in \mathbb{F}_q \)
- An incoming edge represents the side information
- An index code of length \( L \) is
  1. An encoding function \( E : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^L \)
  2. \( n \) decoding functions \( D_i \) s.t. for any \( x \in \mathbb{F}_q^n \)
     \[ D_i(E(x), x_{|N(i)}) = x_i. \]
Index Coding [Birk, Kol 1998]

- Side information graph \( G \)

- \( \text{Index}_q(G) = \text{the shortest length of an index code over } \mathbb{F}_q \)
Index Coding [Birk, Kol 1998]

- Index\(_q(G)\) = the shortest length of an index code over \(\mathbb{F}_q\)
- Index\((G) = \inf_q \text{Index}_q(G)\)

Side information graph \(G\)
Index Coding [Birk, Kol 1998]

- Side information graph $G$

- $\text{Index}_q(G) = \text{the shortest length of an index code over } \mathbb{F}_q$

- $\text{Index}(G) = \inf_q \text{Index}_q(G) = \lim_{q \to \infty} \text{Index}_q(G)$
Index Coding - Continued
Index Coding - Continued

- Index code: an encoding function \( E : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^L \)
Index Coding - Continued

- Index code: an encoding function $E : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^L$
- The confusion of $G$, $\text{Conf}(G)$ has
Index Coding - Continued

- Index code: an encoding function $E : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^L$
- The confusion of $G$, $\text{Conf}(G)$ has
  - Vertex set: $\mathbb{F}_q^n$
Index Coding - Continued

- Index code: an encoding function $E : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^L$
- The confusion of $G$, Conf($G$) has
  - **Vertex set:** $\mathbb{F}_q^n$
  - **Edge set:** $v = (v_1, ..., v_n) \sim u = (u_1, ..., u_n)$ if
Index Coding - Continued

- Index code: an encoding function $E : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^L$

- The confusion of $G$, Conf($G$) has

  - **Vertex set:** $\mathbb{F}_q^n$

  - **Edge set:** $v = (v_1, ..., v_n) \sim u = (u_1, ..., u_n)$ if

    $\exists i \ s.t. v_i \neq u_i$
Index Coding - Continued

- Index code: an encoding function $E : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^L$
- The confusion of $G$, $\text{Conf}(G)$ has
  - **Vertex set:** $\mathbb{F}_q^n$
  - **Edge set:** $v = (v_1, ..., v_n) \sim u = (u_1, ..., u_n)$ if
    \[
    \text{Exists } i \text{ s.t. } v_i \neq u_i \text{ and } v|_{N(i)} = u|_{N(i)}
    \]
Index Coding - Continued

- **Index code**: an encoding function \( E : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^L \)

- The confusion of \( G \), \( \text{Conf}(G) \) has
  - **Vertex set**: \( \mathbb{F}_q^n \)
  - **Edge set**: \( v = (v_1, ..., v_n) \sim u = (u_1, ..., u_n) \) if
    
    \[
    \text{Exists } i \text{ s.t. } v_i \neq u_i \text{ and } v|_{N(i)} = u|_{N(i)}
    \]

- **Fact**: \( E(\cdot) \) is an index code iff it is a **proper** coloring of \( \text{Conf}(G) \)
Index Coding - Continued

- Index code: an encoding function $E : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^L$
- The confusion of $G$, $\text{Conf}(G)$ has

  - **Vertex set**: $\mathbb{F}_q^n$
  - **Edge set**: $v = (v_1, ..., v_n) \sim u = (u_1, ..., u_n)$ if
    
    $\exists i\ s.t. v_i \neq u_i$ and $v|_{N(i)} = u|_{N(i)}$

- **Fact**: $E(\cdot)$ is an index code iff it is a **proper** coloring of $\text{Conf}(G)$

**Corollary**

Given a graph $G$
Index Coding - Continued

- Index code: an encoding function $E : \mathbb{F}_q^n \to \mathbb{F}_q^L$
- The confusion of $G$, $\text{Conf}(G)$ has
  - **Vertex set:** $\mathbb{F}_q^n$
  - **Edge set:** $v = (v_1, \ldots, v_n) \sim u = (u_1, \ldots, u_n)$ if
    
    $\exists i \text{ s.t. } v_i \neq u_i \text{ and } v|_{N(i)} = u|_{N(i)}$

- **Fact:** $E(\cdot)$ is an index code iff it is a *proper* coloring of $\text{Conf}(G)$

**Corollary**

Given a graph $G$

$$\text{Index}_q(G) = \lceil \log_q \chi(\text{Conf}(G)) \rceil,$$

where $\chi(\cdot)$ is the chromatic number
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

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

LRC codes
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
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 $\text{Cap}(G)$
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

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

Proof:
Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

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

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

Proof:

- $\chi_f = \text{fractional chromatic number}$
Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

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

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

Proof:

- $\chi_f =$ fractional chromatic number
- $H_q = \text{Conf}_q(G)$ is a Cayley Graph of $(\mathbb{F}_q^n, +)$
Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

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

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

Proof:

- $\chi_f$ = fractional chromatic number
- $H_q = \text{Conf}_q(G)$ is a Cayley Graph of $(\mathbb{F}_q^n, +)$ → is vertex transitive
Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

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

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

Proof:

- $\chi_f = $ fractional chromatic number
- $H_q = \text{Conf}_q(G)$ is a Cayley Graph of $(\mathbb{F}_q^n, +)$ \rightarrow is vertex transitive

$$\Rightarrow \alpha(H_q) \cdot \chi_f(H_q) = |V(H_q)| = q^n$$
Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

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

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

Proof:

- $\chi_f = \text{fractional chromatic number}$
- $H_q = \text{Conf}_q(G)$ is a Cayley Graph of $(\mathbb{F}_q^n, +) \rightarrow \text{is vertex transitive}$

\[
\Rightarrow \alpha(H_q) \cdot \chi_f(H_q) = |V(H_q)| = q^n \\
\Rightarrow \lim_{q \to \infty} \log_q \alpha(H_q) + \lim_{q \to \infty} \log_q \chi_f(H_q) = n
\]
Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]
Let $G = (V, E), |V| = n$, then

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

Proof:

- $\chi_f = \text{fractional chromatic number}$
- $H_q = \text{Conf}_q(G)$ is a Cayley Graph of $(\mathbb{F}_q^n, +)$ → is vertex transitive

$$\Rightarrow \alpha(H_q) \cdot \chi_f(H_q) = |V(H_q)| = q^n$$
$$\Rightarrow \lim_{q \to \infty} \log_q \alpha(H_q) + \lim_{q \to \infty} \log_q \chi_f(H_q) = n$$
$$\Rightarrow \text{Cap}(G) + \lim_{q \to \infty} \log_q \chi_f(H_q) = n$$
Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

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

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

Proof:

- $\text{Cap}(G) + \lim_{q \to \infty} \log_q \chi_f(H_q) = n$
Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

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

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

Proof:

- $\text{Cap}(G) + \lim_{q \to \infty} \log_q \chi_f(H_q) = n$
- For any graph, and in particular, for $H_q$
Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

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

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

Proof:

- $\text{Cap}(G) + \lim_{q \to \infty} \log_q \chi_f(H_q) = n$

- For any graph, and in particular, for $H_q$

  $$\chi_f(H_q) \leq \chi(H_q) \leq \chi_f(H_q) \cdot \ln |V(H_q)|$$
Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

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

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

Proof:

- $\text{Cap}(G) + \lim_{q \to \infty} \log_q \chi_f(H_q) = n$

- For any graph, and in particular, for $H_q$

  $$\chi_f(H_q) \leq \chi(H_q) \leq \chi_f(H_q) \cdot \ln |V(H_q)|$$

  $$\Rightarrow \lim_{q \to \infty} \log_q \chi_f(H_q) = \lim_{q \to \infty} \log_q \chi(H_q)$$
Duality between Storage Capacity and Index Coding

Theorem [Mazumdar 14, Shanmugam and Dimakis 14]

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

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

Proof:

- $\text{Cap}(G) + \lim_{q \to \infty} \log_q \chi_f(H_q) = n$

- For any graph, and in particular, for $H_q$

  $$\chi_f(H_q) \leq \chi(H_q) \leq \chi_f(H_q) \cdot \ln |V(H_q)|$$

  $$\Rightarrow \lim_{q \to \infty} \log_q \chi_f(H_q) = \lim_{q \to \infty} \log_q \chi(H_q) = \text{Index}(G)$$
Bounds on $\text{Cap}(G)$
Bounds on $\text{Cap}(G)$

- Any bound on $\text{Index}(G) \Rightarrow$ a bound on $\text{Cap}(G)$
Bounds on $\text{Cap}(G)$

- Any bound on $\text{Index}(G) \Rightarrow$ a bound on $\text{Cap}(G)$

Upper Bounds:
Bounds on $\text{Cap}(G)$

- Any bound on $\text{Index}(G)$ $\Rightarrow$ a bound on $\text{Cap}(G)$

Upper Bounds:

- $\text{Cap}(G) \leq n - \alpha(G) = \text{VertexCover}(G)$
Bounds on $\text{Cap}(G)$

- Any bound on $\text{Index}(G)$ $\Rightarrow$ a bound on $\text{Cap}(G)$

**Upper Bounds:**
- $\text{Cap}(G) \leq n - \alpha(G) = \text{VertexCover}(G)$
  - First proof: $\text{Index}(G) \geq \alpha(G)$
Bounds on \( \text{Cap}(G) \)

- Any bound on \( \text{Index}(G) \) ⇒ a bound on \( \text{Cap}(G) \)

**Upper Bounds:**
- \( \text{Cap}(G) \leq n - \alpha(G) = \text{VertexCover}(G) \)
  - First proof: \( \text{Index}(G) \geq \alpha(G) \)
  - Second Proof:
Bounds on $\text{Cap}(G)$

- Any bound on $\text{Index}(G)$ $\Rightarrow$ a bound on $\text{Cap}(G)$

**Upper Bounds:**

- $\text{Cap}(G) \leq n - \alpha(G) = \text{VertexCover}(G)$
  - First proof: $\text{Index}(G) \geq \alpha(G)$
  - Second Proof:
    1. $S \subseteq V(G)$, a maximal independent set
Bounds on $\text{Cap}(G)$

- Any bound on $\text{Index}(G)$ ⇒ a bound on $\text{Cap}(G)$

**Upper Bounds:**
- $\text{Cap}(G) \leq n - \alpha(G) = \text{VertexCover}(G)$
  - First proof: $\text{Index}(G) \geq \alpha(G)$
  - Second Proof:
    1. $S \subseteq V(G)$, a maximal independent set
    2. The information in $S$ can be recovered by the nodes $V(G) \setminus S$
Bounds on $\text{Cap}(G)$

- Any bound on $\text{Index}(G)$ $\Rightarrow$ a bound on $\text{Cap}(G)$

**Upper Bounds:**

- $\text{Cap}(G) \leq n - \alpha(G) = \text{VertexCover}(G)$

- Shannon type inequalities (Recall the proof for the pentagon)
Bounds on $\text{Cap}(G)$

- Any bound on $\text{Index}(G) \Rightarrow$ a bound on $\text{Cap}(G)$

**Upper Bounds:**

- $\text{Cap}(G) \leq n - \alpha(G) = \text{VertexCover}(G)$

- Shannon type inequalities (Recall the proof for the pentagon)

- Non-Shannon type inequalities
Bounds on $\text{Cap}(G)$

- Any bound on $\text{Index}(G) \Rightarrow$ a bound on $\text{Cap}(G)$

**Upper Bounds:**

- $\text{Cap}(G) \leq n - \alpha(G) = \text{VertexCover}(G)$

- Shannon type inequalities (Recall the proof for the pentagon)

- Non-Shannon type inequalities
  - Theorem [Baber, Chistofides, Dang, Riis, Vaughan]:
    There exists a graph $G$ on 10 vertices s.t.
Bounds on $\text{Cap}(G)$

- Any bound on $\text{Index}(G)$ $\Rightarrow$ a bound on $\text{Cap}(G)$

**Upper Bounds:**

- $\text{Cap}(G) \leq n - \alpha(G) = \text{VertexCover}(G)$

- Shannon type inequalities (Recall the proof for the pentagon)

- Non-Shannon type inequalities
  - Theorem [Baber, Chistofides, Dang, Riis, Vaughan]:
    There exists a graph $G$ on 10 vertices s.t.
      - Shannon type inequalities give: $\text{Cap}(G) \leq 114/17 = 6.705…$
Bounds on $\text{Cap}(G)$

- Any bound on $\text{Index}(G) \Rightarrow$ a bound on $\text{Cap}(G)$

**Upper Bounds:**

- $\text{Cap}(G) \leq n - \alpha(G) = \text{VertexCover}(G)$

- Shannon type inequalities (Recall the proof for the pentagon)

- Non-Shannon type inequalities
  - Theorem [Baber, Chistofides, Dang, Riis, Vaughan]: There exists a graph $G$ on 10 vertices s.t.
    - Shannon type inequalities give: $\text{Cap}(G) \leq 114/17 = 6.705…$
    - Non-Shannon type inequalities give: $\text{Cap}(G) \leq 20/3 = 6.666…$
Bounds on $\text{Cap}(G)$

- Any bound on $\text{Index}(G) \Rightarrow$ a bound on $\text{Cap}(G)$

**Upper Bounds:**

- $\text{Cap}(G) \leq n - \alpha(G) = \text{VertexCover}(G)$

- Shannon type inequalities (Recall the proof for the pentagon)

- Non-Shannon type inequalities
  - Theorem [Baber, Chistofides, Dang, Riis, Vaughan]: There exists a graph $G$ on 10 vertices s.t.
    - Shannon type inequalities give: $\text{Cap}(G) \leq 114/17 = 6.705\ldots$
    - Non-Shannon type inequalities give: $\text{Cap}(G) \leq 20/3 = 6.666\ldots$

**Lower bounds:**
Bounds on $\text{Cap}(G)$

- Any bound on $\text{Index}(G) \Rightarrow$ a bound on $\text{Cap}(G)$

**Upper Bounds:**

- $\text{Cap}(G) \leq n - \alpha(G) = \text{VertexCover}(G)$

- Shannon type inequalities (Recall the proof for the pentagon)

- Non-Shannon type inequalities
  - Theorem [Baber, Chistofides, Dang, Riis, Vaughan]:
    There exists a graph $G$ on 10 vertices s.t.
    - Shannon type inequalities give: $\text{Cap}(G) \leq 114/17 = 6.705…$
    - Non-Shannon type inequalities give: $\text{Cap}(G) \leq 20/3 = 6.666…$

**Lower bounds:**

- $\text{Index}(G) \leq \min\{\chi_f(\overline{G}), \text{Minrank}(G)\}$
Bounds on $\text{Cap}(G)$

- Any bound on $\text{Index}(G) \Rightarrow$ a bound on $\text{Cap}(G)$

Upper Bounds:

- $\text{Cap}(G) \leq n - \alpha(G) = \text{VertexCover}(G)$

- Shannon type inequalities (Recall the proof for the pentagon)

- Non-Shannon type inequalities
  - Theorem [Baber, Chistofides, Dang, Riis, Vaughan]:
    There exists a graph $G$ on 10 vertices s.t.
    - Shannon type inequalities give: $\text{Cap}(G) \leq 114/17 = 6.705…$
    - Non-Shannon type inequalities give: $\text{Cap}(G) \leq 20/3 = 6.666…$

Lower bounds:

- $\text{Index}(G) \leq \min\{\chi_f(\overline{G}), \text{Minrank}(G)\}$

- $\text{Cap}(G) \geq n - \min\{\chi_f(\overline{G}), \text{Minrank}(G)\}$
Guessing game on graphs [Riis2005]
Guessing game on graphs [Riis2005]

- $n$ players
Guessing game on graphs [Riis2005]

- $n$ players
- Red/Blue hat is assigned independently to each player. $x_j = \begin{cases} 0 & \text{if blue} \\ 1 & \text{if red} \end{cases}$
Guessing game on graphs [Riis2005]

- $n$ players

- Red/Blue hat is assigned independently to each player. $x_j = \begin{cases} 0 & \text{if blue} \\ 1 & \text{if red} \end{cases}$

- Each player:
Guessing game on graphs [Riis2005]

- $n$ players
- Red/Blue hat is assigned independently to each player. $x_j = \begin{cases} 0 & \text{if blue} \\ 1 & \text{if red} \end{cases}$
- Each player:
  1. observes **only** the hats of his mates (complete graph)
Guessing game on graphs [Riis2005]

- $n$ players

- Red/Blue hat is assigned independently to each player. $x_j = \begin{cases} 0 & \text{if blue} \\ 1 & \text{if red} \end{cases}$

- Each player:
  1. observes only the hats of his mates (complete graph)
  2. makes a guess on the color of his hat
Guessing game on graphs [Riis2005]

- $n$ players

- Red/Blue hat is assigned independently to each player. $x_j = \begin{cases} 0 & \text{if blue} \\ 1 & \text{if red} \end{cases}$

- Each player:
  1. observes only the hats of his mates (complete graph)
  2. makes a guess on the color of his hat

- What is the probability that all players guessed correctly?
Guessing game on graphs [Riis2005]

- $n$ players

- Red/Blue hat is assigned independently to each player. $x_j = \begin{cases} 0 & \text{if blue} \\ 1 & \text{if red} \end{cases}$

- Each player:
  1. observes only the hats of his mates (complete graph)
  2. makes a guess on the color of his hat

- What is the probability that all players guessed correctly?

- Random guess $\rightarrow P_{\text{success}} = \left(\frac{1}{2}\right)^n$
Guessing game on graphs [Riis2005]

• $n$ players

• Red/Blue hat is assigned independently to each player. $x_j = \begin{cases} 0 & \text{if blue} \\ 1 & \text{if red} \end{cases}$

• Each player:
  1. observes only the hats of his mates (complete graph)
  2. makes a guess on the color of his hat

• What is the probability that all players guessed correctly?

• Random guess $\rightarrow P_{\text{success}} = \left(\frac{1}{2}\right)^n$

• Strategy: assume $\sum_{j=1}^{n} x_j = 0$
Guessing game on graphs [Riis2005]

- $n$ players
- Red/Blue hat is assigned independently to each player. $x_j = \begin{cases} 0 & \text{if blue} \\ 1 & \text{if red} \end{cases}$
- Each player:
  1. observes **only** the hats of his mates (complete graph)
  2. makes a guess on the color of his hat
- What is the probability that **all** players guessed correctly?
- Random guess $\rightarrow P_{\text{success}} = \left(\frac{1}{2}\right)^n$
- Strategy: assume $\sum_{j=1}^n x_j = 0 \rightarrow P_{\text{success}} = \frac{1}{2}$
Guessing game on graphs [Riis2005]

- Arbitrary graphs?
Guessing game on graphs [Riis2005]

- Arbitrary graphs?
- Neighbors $A_1 = \{2\}$, $A_2 = \{1, 3\}$, $A_3 = \{1\}$
Guessing game on graphs [Riis2005]

- Arbitrary graphs?
- Neighbors $A_1 = \{2\}$, $A_2 = \{1, 3\}$, $A_3 = \{1\}$
- Fundamental limit on the amount of improvement over random guesses?
Guessing game on graphs [Riis2005]

- Arbitrary graphs?
- Neighbors $A_1 = \{2\}$, $A_2 = \{1, 3\}$, $A_3 = \{1\}$
- Fundamental limit on the amount of improvement over random guesses?
- What is the optimal strategy?
• Arbitrary graphs?

• Neighbors \( A_1 = \{2\}, A_2 = \{1, 3\}, A_3 = \{1\} \)

• Fundamental limit on the amount of improvement over random guesses?

• What is the optimal strategy?

• Connection to index coding [Yi–Sun–Jafar–Gesbert 2015]
Guessing Number [Riis2005]

- Given a (directed) graph $G = (V, E)$, $|V| = n$
Guessing Number [Riis2005]

- Given a (directed) graph $G = (V, E)$, $|V| = n$
- Each player has a hat of $q$ possible colors
Guessing Number [Riis2005]

- Given a (directed) graph $G = (V, E)$, $|V| = n$
- Each player has a hat of $q$ possible colors
- Let $X = (x_1, ..., x_n), x_i \in [q]$ be a hat assignment
Guessing Number [Riis2005]

• Given a (directed) graph \( G = (V, E) \), \(|V| = n\)

• Each player has a hat of \( q \) possible colors

• Let \( X = (x_1, ..., x_n), x_i \in [q] \) be a hat assignment

• A strategy \( f = (f_1, ..., f_n) \) composed of \( n \) functions \( f_i : (x_j : j \in N(i)) \rightarrow [q] \)
Guessing Number [Riis2005]

- Given a (directed) graph $G = (V, E)$, $|V| = n$
- Each player has a hat of $q$ possible colors
- Let $X = (x_1, ..., x_n), x_i \in [q]$ be a hat assignment
- A strategy $f = (f_1, ..., f_n)$ composed of $n$ functions $f_i : (x_j : j \in N(i)) \rightarrow [q]$
- The strategy is successful iff
Guessing Number [Riis2005]

- Given a (directed) graph $G = (V, E)$, $|V| = n$
- Each player has a hat of $q$ possible colors
- Let $X = (x_1, ..., x_n), x_i \in [q]$ be a hat assignment
- A strategy $f = (f_1, ..., f_n)$ composed of $n$ functions $f_i : (x_j : j \in N(i)) \rightarrow [q]$
- The strategy is successful iff
  \[ f_i(x_j : j \in N(i)) = x_i \text{ for all } i \]
Guessing Number [Riis2005]

- Given a (directed) graph \( G = (V, E), |V| = n \)
- Each player has a hat of \( q \) possible colors
- Let \( X = (x_1, ..., x_n), x_i \in [q] \) be a hat assignment
- A strategy \( f = (f_1, ..., f_n) \) composed of \( n \) functions \( f_i : (x_j : j \in N(i)) \to [q] \)
- The strategy is successful iff
  \[
  f_i(x_j : j \in N(i)) = x_i \text{ for all } i
  \]
- Equivalently: The hat assignment \( X \) is a fixed point for the mapping
  \[
  X = (x_1, ..., x_n)
  \]
Guessing Number [Riis2005]

- Given a (directed) graph \( G = (V, E), |V| = n \)
- Each player has a hat of \( q \) possible colors
- Let \( X = (x_1, \ldots, x_n), x_i \in [q] \) be a hat assignment
- A strategy \( f = (f_1, \ldots, f_n) \) composed of \( n \) functions \( f_i : (x_j : j \in N(i)) \to [q] \)
- The strategy is successful iff
  \[
  f_i(x_j : j \in N(i)) = x_i \text{ for all } i
  \]
- Equivalently: The hat assignment \( X \) is a fixed point for the mapping
  \[
  X = (x_1, \ldots, x_n) \mapsto f(X) = (f_1(X), \ldots, f_n(X))
  \]
  \( f_i(X) \) is a function of the hat colors of the neighbors of \( i \)
$q$-Guessing Number - Definition

The $q$-guessing number of $G$

$$\text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q \frac{P(f \text{ is successful})}{P_{\text{rand}}}$$
**Guessing Number** [Riis2005]

$q$-Guessing Number - Definition

The $q$-guessing number of $G$

$$\text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q \frac{P(f \text{ is successful})}{P_{\text{rand}}}$$

- For a fixed strategy $f$
**q-Guessing Number - Definition**

The $q$-guessing number of $G$

\[
\text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q \frac{P(f \text{ is successful})}{P_{\text{rand}}}
\]

- For a fixed strategy $f$

\[
\log_q \frac{P(f \text{ is successful})}{P_{\text{rand}}} = \log_q P(f \text{ is successful}) + n
\]
Guessing Number [Riis2005]

$q$-Guessing Number - Definition

The $q$-guessing number of $G$

$$\text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q \frac{P(f \text{ is successful})}{P_{\text{rand}}}$$

- For a fixed strategy $f$

$$\log_q \frac{P(f \text{ is successful})}{P_{\text{rand}}} = \log_q P(f \text{ is successful}) + n$$

$$= \log_q \frac{|\text{fixed points of } f|}{q^n} + n$$
**q-Guessing Number - Definition**

The $q$-guessing number of $G$

$$\text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q \frac{P(f \text{ is successful})}{P_{\text{rand}}}$$

- For a fixed strategy $f$

$$\log_q \frac{P(f \text{ is successful})}{P_{\text{rand}}} = \log_q P(f \text{ is successful}) + n$$

$$= \log_q \left| \text{fixed points of } f \right| \frac{1}{q^n} + n$$

$$= \log_q \left| \text{fixed points of } f \right|$$
**Guessing Number** [Riis2005]

**q-Guessing Number - Definition**

The $q$-guessing number of $G$

\[
\text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q \frac{P(f \text{ is successful})}{P_{\text{rand}}} = \max_{f \text{ is a strategy}} \log_q |\text{fixed points of } f|
\]

- For a fixed strategy $f$
  \[
  \log_q \frac{P(f \text{ is successful})}{P_{\text{rand}}} = \log_q P(f \text{ is successful}) + n
  = \log_q \frac{|\text{fixed points of } f|}{q^n} + n
  = \log_q |\text{fixed points of } f|
  \]
Guessing Number [Riis2005]

$q$-Guessing Number - Definition

The $q$-guessing number of $G$

$$\text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q \frac{P(f \text{ is successful})}{P_{\text{rand}}} = \max_{f \text{ is a strategy}} \log_q |\text{fixed points of } f|$$
**Guessing Number** [Riis2005]

### $q$-Guessing Number - Definition

The $q$-guessing number of $G$

$$
\text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q \frac{P(f \text{ is successful})}{P_{\text{rand}}} = \max_{f \text{ is a strategy}} \log_q |\text{fixed points of } f|
$$

### Guessing Number - Definition

The guessing number of $G$

$$
\text{Guess}(G) = \sup_{q} \text{Guess}_q(G)
$$
Guessing games and LRC codes on graphs
Guessing games and LRC codes on graphs

- Given a graph $G$
Guessing games and LRC codes on graphs

- Given a graph $G$

$$
\text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q |\text{fixed points of } f|
$$
Guessing games and LRC codes on graphs

- Given a graph $G$

\[
\text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q |\text{fixed points of } f| \\
\text{Cap}_q(G) = \max_{C \subseteq \mathbb{F}_q^n \text{ is a storage code for } G} \log_q |C|
\]
Guessing games and LRC codes on graphs

- Given a graph $G$

  \[
  \text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q |\text{fixed points of } f| \\
  \text{Cap}_q(G) = \max_{C \subseteq \mathbb{F}_q^n \text{ is a storage code for } G} \log_q |C|
  \]

- The set of fixed points of a strategy forms a storage code $C$ for $G$
Guessing games and LRC codes on graphs

- Given a graph $G$

  \[
  \text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q |\text{fixed points of } f| \\
  \text{Cap}_q(G) = \max_{C \subseteq \mathbb{F}_q^n \text{ is a storage code for } G} \log_q |C|
  \]

- The set of fixed points of a strategy forms a storage code $C$ for $G$

  \[
  \text{Guess}_q(G) \leq \text{Cap}_q(G)
  \]
Guessing games and LRC codes on graphs

- Given a graph $G$

  \[
  \text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q |\text{fixed points of } f| \\
  \text{Cap}_q(G) = \max_{C \subseteq \mathbb{F}_q^n \text{ is a storage code for } G} \log_q |C|
  \]

- The set of fixed points of a strategy forms a storage code $C$ for $G$

  \[
  \text{Guess}_q(G) \leq \text{Cap}_q(G)
  \]

- The codewords of a storage code form the set of fixed points of some strategy for the hat game
Guessing games and LRC codes on graphs

- Given a graph $G$

  $$\text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q |\text{fixed points of } f|$$

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

- The set of fixed points of a strategy forms a storage code $C$ for $G$

  $$\text{Guess}_q(G) \leq \text{Cap}_q(G)$$

- The codewords of a storage code form the set of fixed points of some strategy for the hat game

  $$\text{Cap}_q(G) \leq \text{Guess}_q(G)$$
Guessing games and LRC codes on graphs

- Given a graph $G$

$$\text{Guess}_q(G) = \max_{\text{f is a strategy}} \log_q |\text{fixed points of } f|$$

$$\text{Cap}_q(G) = \max_{C \subseteq F_q^n \text{ is a storage code for } G} \log_q |C|$$

- The set of fixed points of a strategy forms a storage code $C$ for $G$

$$\text{Guess}_q(G) \leq \text{Cap}_q(G)$$

- The codewords of a storage code form the set of fixed points of some strategy for the hat game

$$\text{Cap}_q(G) \leq \text{Guess}_q(G)$$

- $\text{Guess}_q(G) = \text{Cap}_q(G)$
Guessing games and LRC codes on graphs

- Given a graph $G$

  \[
  \text{Guess}_q(G) = \max_{f \text{ is a strategy}} \log_q |\text{fixed points of } f| \\
  \text{Cap}_q(G) = \max_{C \subseteq \mathbb{F}_q^n \text{ is a storage code for } G} \log_q |C|
  \]

- The set of fixed points of a strategy forms a storage code $C$ for $G$

  \[
  \text{Guess}_q(G) \leq \text{Cap}_q(G)
  \]

- The codewords of a storage code form the set of fixed points of some strategy for the hat game

  \[
  \text{Cap}_q(G) \leq \text{Guess}_q(G)
  \]

- $\text{Guess}_q(G) = \text{Cap}_q(G)$

- $\text{Guess}(G) = \text{Cap}(G)$
Open questions
Open questions

- Derive improved bounds on $\text{Cap}(G)$
Open questions

- Derive improved bounds on $\text{Cap}(G)$
- How expansion properties of the graph (expander graphs) affect the storage capacity?
Open questions

- Derive improved bounds on $\text{Cap}(G)$
- How expansion properties of the graph (expander graphs) affect the storage capacity?
- Approximate the storage capacity using spectral properties of the graph
Open questions

- Derive improved bounds on $\text{Cap}(G)$
- How expansion properties of the graph (expander graphs) affect the storage capacity?
- Approximate the storage capacity using spectral properties of the graph
- Given a graph $G$, characterize the tradeoff between the distance and the rate of the code
Open questions

- Derive improved bounds on $\text{Cap}(G)$
- How expansion properties of the graph (expander graphs) affect the storage capacity?
- Approximate the storage capacity using spectral properties of the graph
- Given a graph $G$, characterize the tradeoff between the distance and the rate of the code
- Networks with time varying topologies: Develop algorithms that efficiently shift between storage codes for different topologies