Keywords:

Error-correcting codes,

low-degree polynomials,

randomness, linear algebra

questions *begging* to be resolved

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f, i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0,1\}^m$.

1	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f, i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0,1\}^m$.

1	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1

Can you recover f?

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f, i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0,1\}^m$.

1	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1

Can you recover f?

Of course! Just interpolate.

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f, i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0,1\}^m$. But she corrupts 49% of the bits.

1	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1

Can you recover f?

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f, i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0,1\}^m$. But she corrupts 49% of the bits.

1	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1

Can you recover f?

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f, i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0,1\}^m$. But she corrupts 49% of the bits.

0	1	1	0	1	0	0	1	1	0	0	0	1	1	1	1

Can you recover f?

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f, i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0,1\}^m$. But she corrupts 49% of the bits.

0	1	1	0	1	0	0	1	1	0	0	0	1	1	1	1

Can you recover f?

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f, i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0,1\}^m$. But she corrupts 49% of the bits.

0	1	1	0	1	0	0	1	1	0	0	0	1	1	1	1

Can you recover f?

Impossible if errors are adversarial...

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f, i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0,1\}^m$. But she corrupts 49% of the bits randomly.

0	1	1	0	1	0	0	1	1	0	0	0	1	1	1	1

Can you recover f?

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f, i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0,1\}^m$. But she corrupts 49% of the bits randomly.

0	1	1	0	1	0	0	1	1	0	0	0	1	1	1	1

Can you recover f?

This talk: Yes, and we can do it efficiently!

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f, i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0,1\}^m$. But she corrupts 49% of the bits randomly.

0	1	1	0	1	0	0	1	1	0	0	0	1	1	1	1

Can you recover f?

This talk: Yes, and we can do it efficiently!

("Efficiently decoding Reed-Muller codes from random errors")

Efficiently decoding Reed-Muller codes from random errors

Ramprasad Saptharishi
TIFR

Amir Shpilka
Tel Aviv University

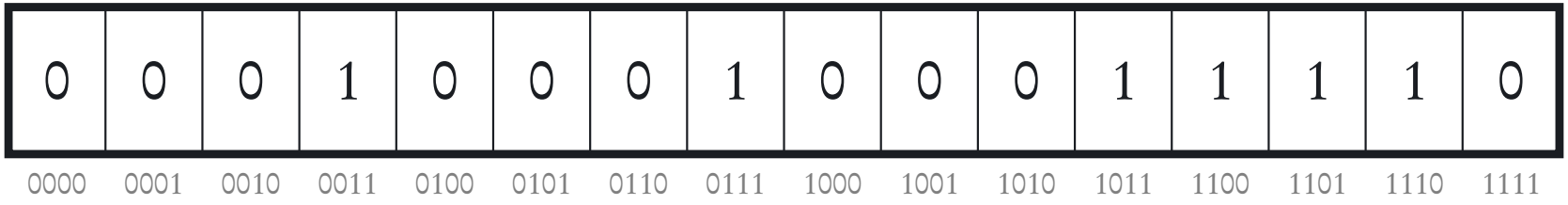
Ben Lee Volk
Tel Aviv University

Reed-Muller Codes: $RM(m, r)$

Message: A degree polynomial $f \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree at most r.

Encoding: The evaluation of f on all points in $\{0,1\}^m$.

$$f(x_1, x_2, x_3, x_4) = x_1x_2 + x_3x_4$$

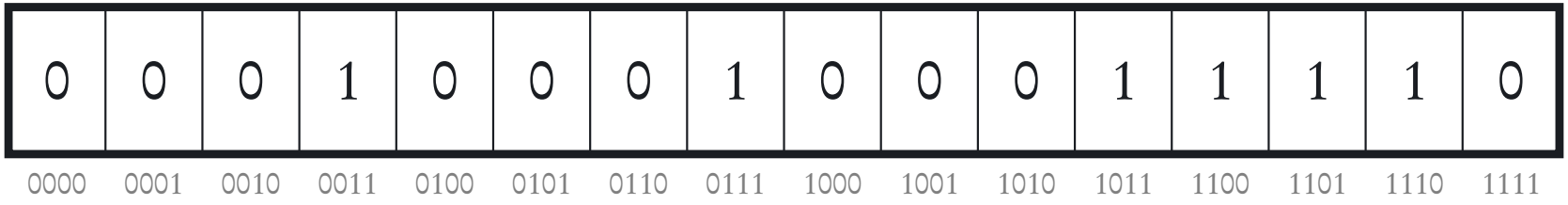


Reed-Muller Codes: $RM(m, r)$

Message: A degree polynomial $f \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree at most r.

Encoding: The evaluation of f on all points in $\{0,1\}^m$.

$$f(x_1, x_2, x_3, x_4) = x_1x_2 + x_3x_4$$



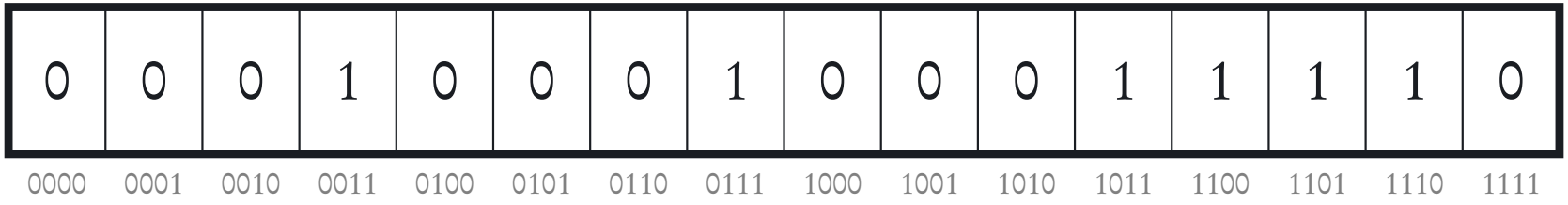
A linear code, with

Reed-Muller Codes: $RM(m, r)$

Message: A degree polynomial $f \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree at most r.

Encoding: The evaluation of f on all points in $\{0,1\}^m$.

$$f(x_1, x_2, x_3, x_4) = x_1x_2 + x_3x_4$$



A linear code, with

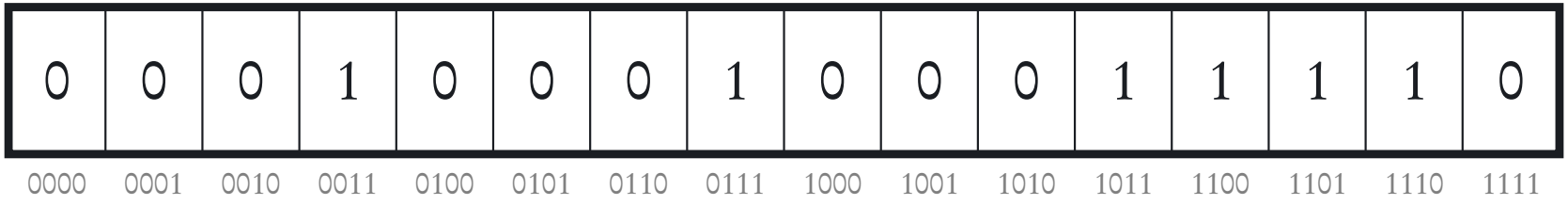
- Block Length: $2^m := n$

Reed-Muller Codes: $RM(m, r)$

Message: A degree polynomial $f \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree at most r.

Encoding: The evaluation of f on all points in $\{0,1\}^m$.

$$f(x_1, x_2, x_3, x_4) = x_1x_2 + x_3x_4$$



A linear code, with

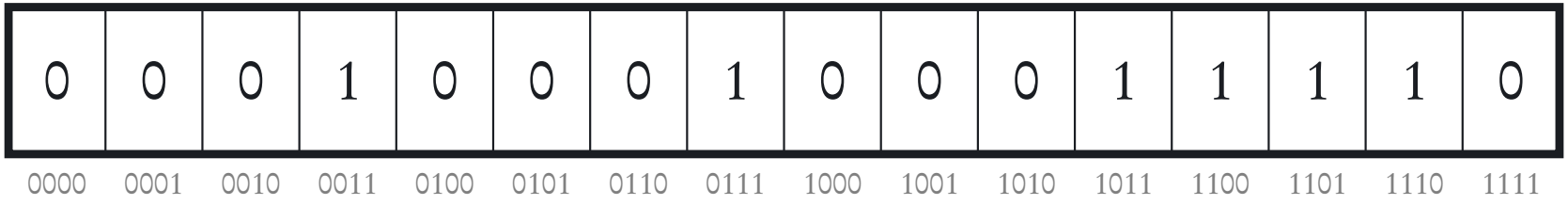
- Block Length: $2^m := n$
- Distance: 2^{m-r} (lightest codeword: $x_1x_2 \cdots x_r$)

Reed-Muller Codes: $RM(m, r)$

Message: A degree polynomial $f \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree at most r.

Encoding: The evaluation of f on all points in $\{0,1\}^m$.

$$f(x_1, x_2, x_3, x_4) = x_1x_2 + x_3x_4$$



A linear code, with

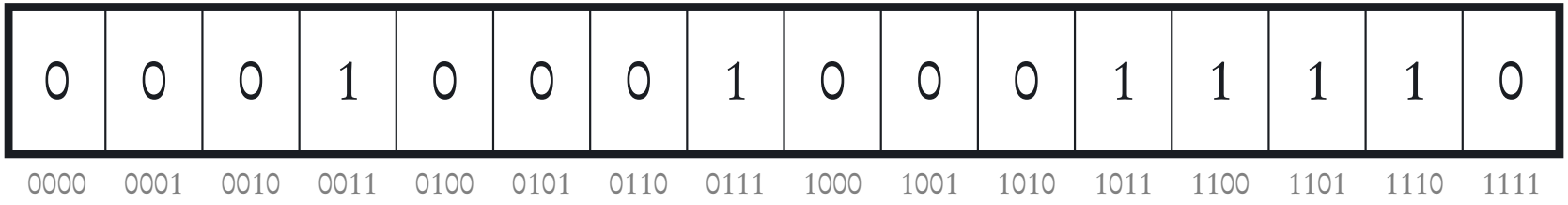
- Block Length: $2^m := n$
- Distance: 2^{m-r} (lightest codeword: $x_1x_2 \cdots x_r$)
- Dimension: $\binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{r} =: \binom{m}{\leq r}$

Reed-Muller Codes: $RM(m, r)$

Message: A degree polynomial $f \in \mathbb{F}_2[x_1, \cdots, x_m]$ of degree at most r.

Encoding: The evaluation of f on all points in $\{0,1\}^m$.

$$f(x_1, x_2, x_3, x_4) = x_1x_2 + x_3x_4$$

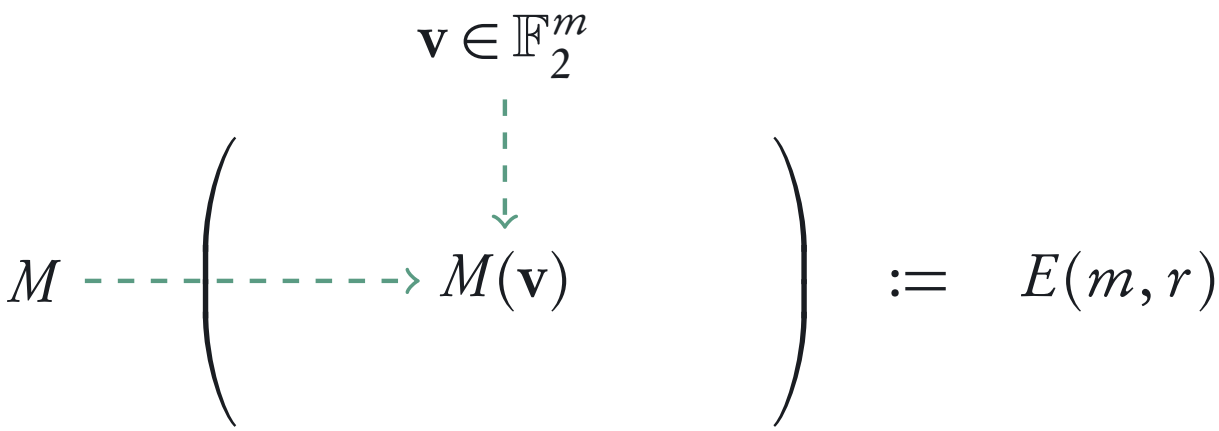


A linear code, with

- Block Length: $2^m := n$
- Distance: 2^{m-r} (lightest codeword: $x_1x_2 \cdots x_r$)
- Dimension: $\binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{r} =: \binom{m}{\leq r}$
- Rate: dimension/block length $= \binom{m}{\leq r}/2^m$

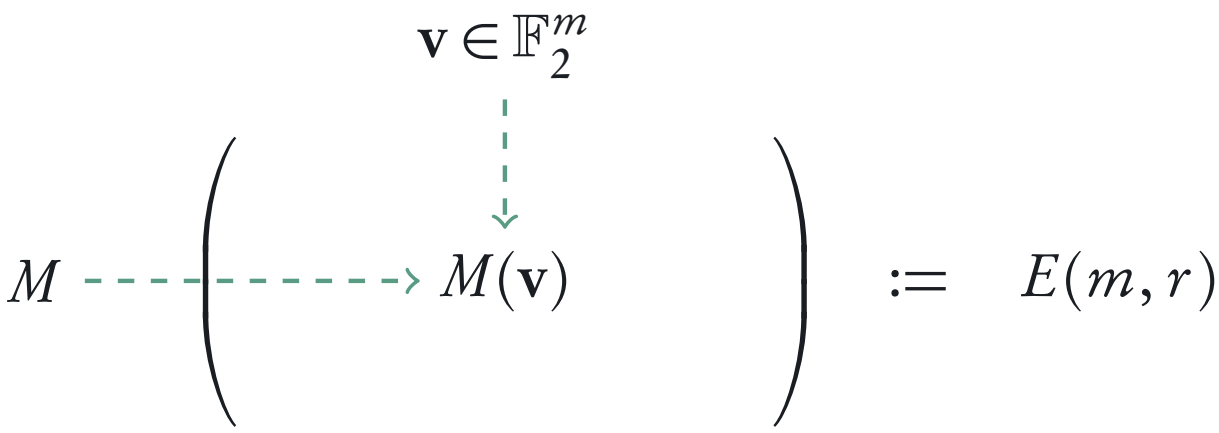
Reed-Muller Codes: $RM(m, r)$

Generator Matrix: evaluation matrix of $\deg \leq r$ monomials:



Reed-Muller Codes: $RM(m, r)$

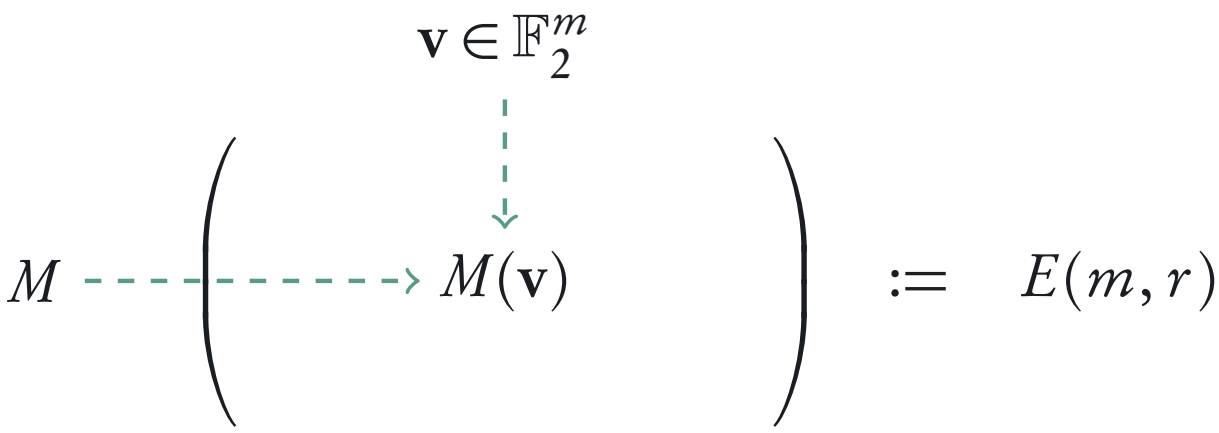
Generator Matrix: evaluation matrix of $\deg \leq r$ monomials:



(Every codeword is spanned by the rows.)

Reed-Muller Codes: $RM(m, r)$

Generator Matrix: evaluation matrix of $\deg \leq r$ monomials:



(Every codeword is spanned by the rows.)

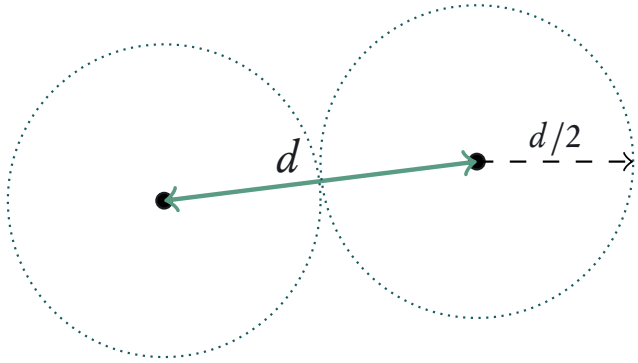
Also called the *inclusion matrix*
($M(\mathbf{v}) = 1$ if and only if "$M \subset \mathbf{v}$").

Decoding $RM(m, r)$

Worst Case Errors:

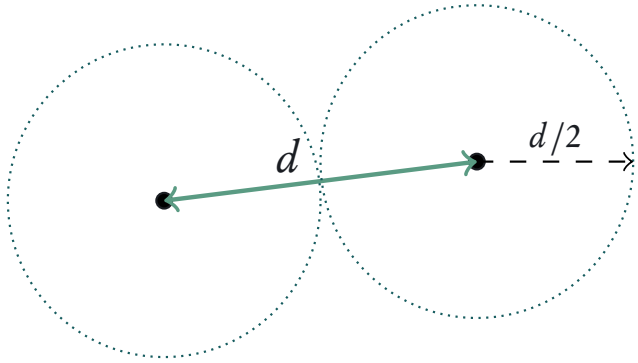
Decoding $RM(m, r)$

Worst Case Errors: Up to $d/2$ ($d = 2^{m-r}$ is minimal distance).



Decoding $RM(m, r)$

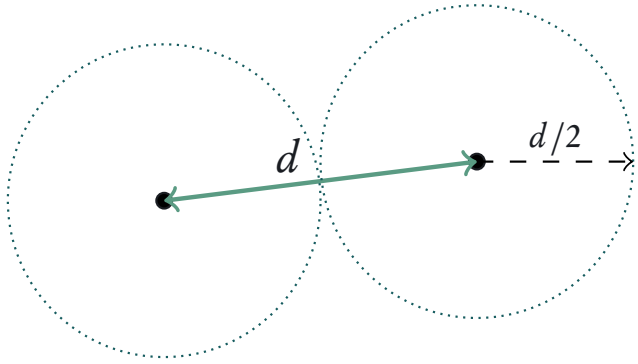
Worst Case Errors: Up to $d/2$ ($d = 2^{m-r}$ is minimal distance).



(algorithm by [Reed54])

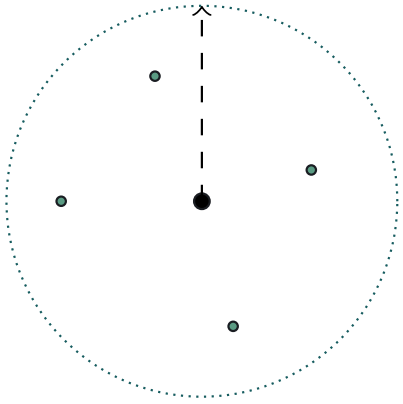
Decoding $RM(m, r)$

Worst Case Errors: Up to $d/2$ ($d = 2^{m-r}$ is minimal distance).



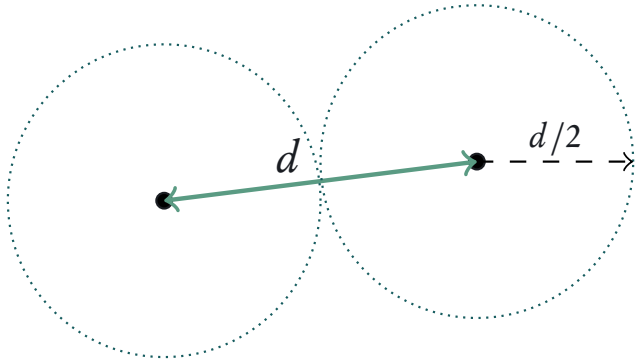
(algorithm by [Reed54])

List Decoding: max radius with constant # of words



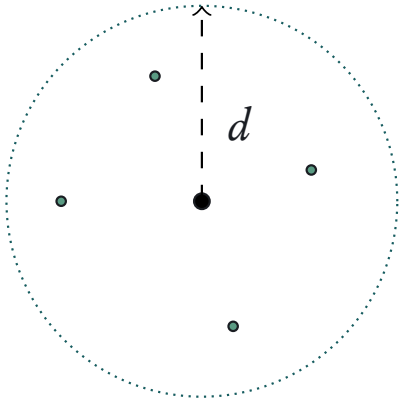
Decoding $RM(m, r)$

Worst Case Errors: Up to $d/2$ ($d = 2^{m-r}$ is minimal distance).



(algorithm by [Reed54])

List Decoding: max radius with constant # of words



[Gopalan-Klivans-Zuckerman08, Bhowmick-Lovett15]

List decoding radius = d.

Two schools of study

Two schools of study

- **Hamming Model** a.k.a worst-case errors
  - Generally the model of interest for complexity theorists,

Two schools of study

- **Hamming Model** a.k.a worst-case errors
  - Generally the model of interest for complexity theorists,
  - Reed-Muller codes are not the best for these (far from optimal rate-distance tradeoffs).

Two schools of study

- **Hamming Model** a.k.a worst-case errors
  - Generally the model of interest for complexity theorists,
  - Reed-Muller codes are not the best for these (far from optimal rate-distance tradeoffs).
- **Shannon Model** a.k.a random errors
  - The standard model for coding theorists,

Two schools of study

- **Hamming Model** a.k.a worst-case errors
  - Generally the model of interest for complexity theorists,
  - Reed-Muller codes are not the best for these (far from optimal rate-distance tradeoffs).
- **Shannon Model** a.k.a random errors
  - The standard model for coding theorists,
  - Recent breakthroughs (e.g. Arıkan's polar codes),

Two schools of study

- **Hamming Model** a.k.a worst-case errors
  - Generally the model of interest for complexity theorists,
  - Reed-Muller codes are not the best for these (far from optimal rate-distance tradeoffs).
- **Shannon Model** a.k.a random errors
  - The standard model for coding theorists,
  - Recent breakthroughs (e.g. Arıkan's polar codes),
  - An ongoing research endeavor:
    How do Reed-Muller codes perform in the Shannon model?

Models for random corruptions (channels)

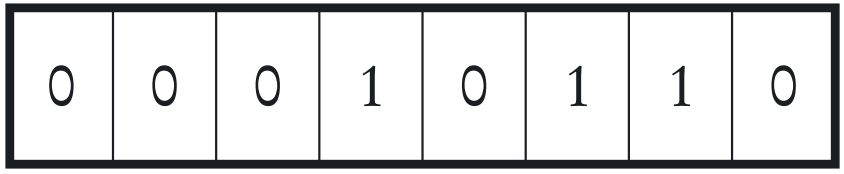
Binary Erasure Channel — BEC(p)

Each bit independently replaced by '?' with probability p

Models for random corruptions (channels)

Binary Erasure Channel — BEC(p)

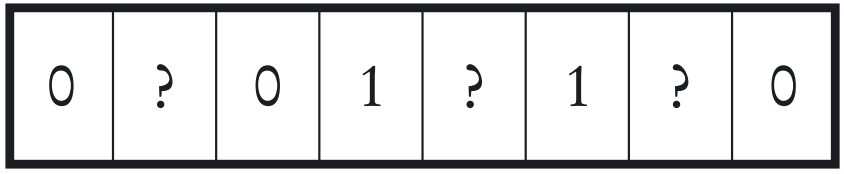
Each bit independently replaced by '?' with probability p



Models for random corruptions (channels)

Binary Erasure Channel — BEC(p)

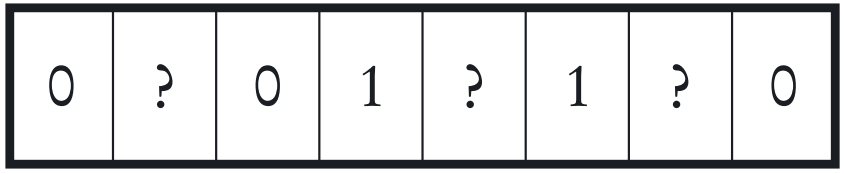
Each bit independently replaced by '?' with probability p



Models for random corruptions (channels)

Binary Erasure Channel — BEC(p)

Each bit independently replaced by '?' with probability p



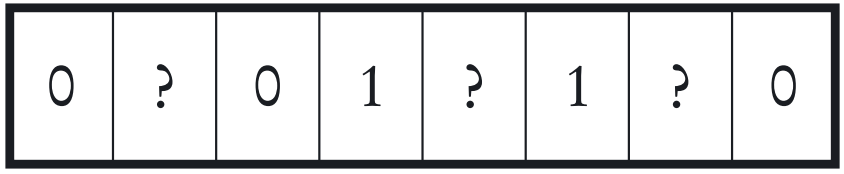
Binary Symmetric Channel — BSC(p)

Each bit independently flipped with probability p

Models for random corruptions (channels)

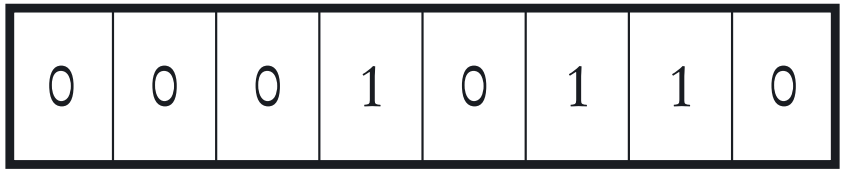
Binary Erasure Channel — BEC(p)

Each bit independently replaced by '?' with probability p



Binary Symmetric Channel — BSC(p)

Each bit independently flipped with probability p



Models for random corruptions (channels)

Binary Erasure Channel — BEC(p)

Each bit independently replaced by '?' with probability p

0	?	0	1	?	1	?	0

Binary Symmetric Channel — BSC(p)

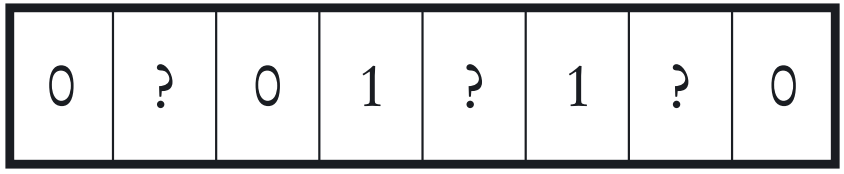
Each bit independently flipped with probability p

0	1	0	1	1	1	0	0

Models for random corruptions (channels)

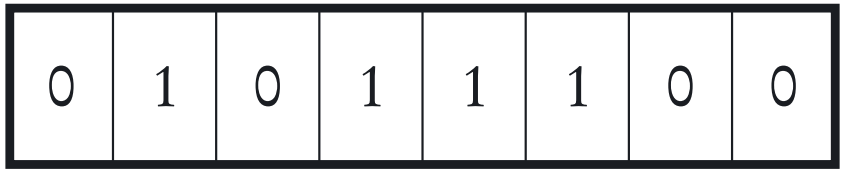
Binary Erasure Channel — BEC(p)

Each bit independently replaced by '?' with probability p



Binary Symmetric Channel — BSC(p)

Each bit independently flipped with probability p



(almost) equiv: fixed number $t \approx pn$ of random errors

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

Binary Erasure Channel — BEC(p)

Each bit independently replaced by '?' with probability p

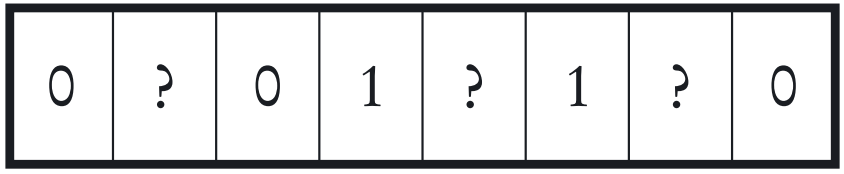
0	?	0	1	?	1	?	0

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

Binary Erasure Channel — BEC(p)

Each bit independently replaced by '?' with probability p



If X^n is transmitted to the channel and received as Y^n, how many bits of information about X^n do we get from Y^n?

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

Binary Erasure Channel — BEC(p)

Each bit independently replaced by '?' with probability p

0	?	0	1	?	1	?	0

If X^n is transmitted to the channel and received as Y^n, how many bits of information about X^n do we get from Y^n?

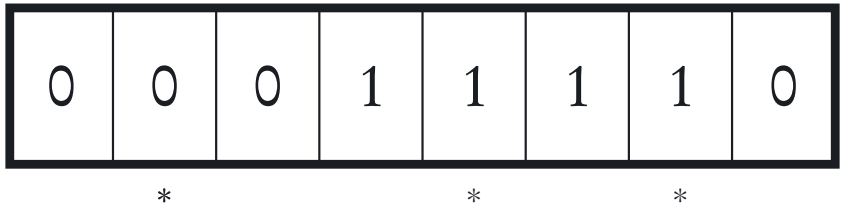
Intuitively, $(1-p)n$.

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

Binary Symmetric Channel — BSC(p)

Each bit independently flipped with probability p

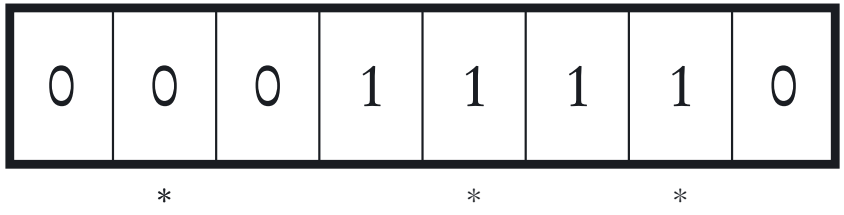


Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

Binary Symmetric Channel — BSC(p)

Each bit independently flipped with probability p



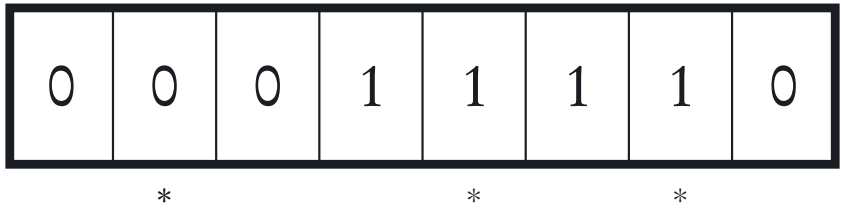
If X^n is transmitted to the channel and received as Y^n, how many bits of information about X^n do we get from Y^n?

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

Binary Symmetric Channel — BSC(p)

Each bit independently flipped with probability p



If X^n is transmitted to the channel and received as Y^n, how many bits of information about X^n do we get from Y^n?

Intuitively, $(1-H(p))n$. (as $\binom{n}{pn} \approx 2^{H(p)\cdot n}$)

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

[Shannon48] Maximum rate that enables decoding (w.h.p.) is:

$$\begin{array}{ll} 1-p & \text{for BEC}(p), \\ 1-H(p) & \text{for BSC}(p). \end{array}$$

Codes achieving this bound called capacity achieving.

Category:Capacity-achieving codes

Help

From Wikipedia, the free encyclopedia

Pages in category "Capacity-achieving codes"

This category contains only the following page. This list may not reflect recent changes (learn more).

P

- Polar code (coding theory)

Categories: Error detection and correction

Motivating questions for this talk

How well does Reed-Muller codes perform in the Shannon Model?

In BEC(p)?

In BSC(p)?

Motivating questions for this talk

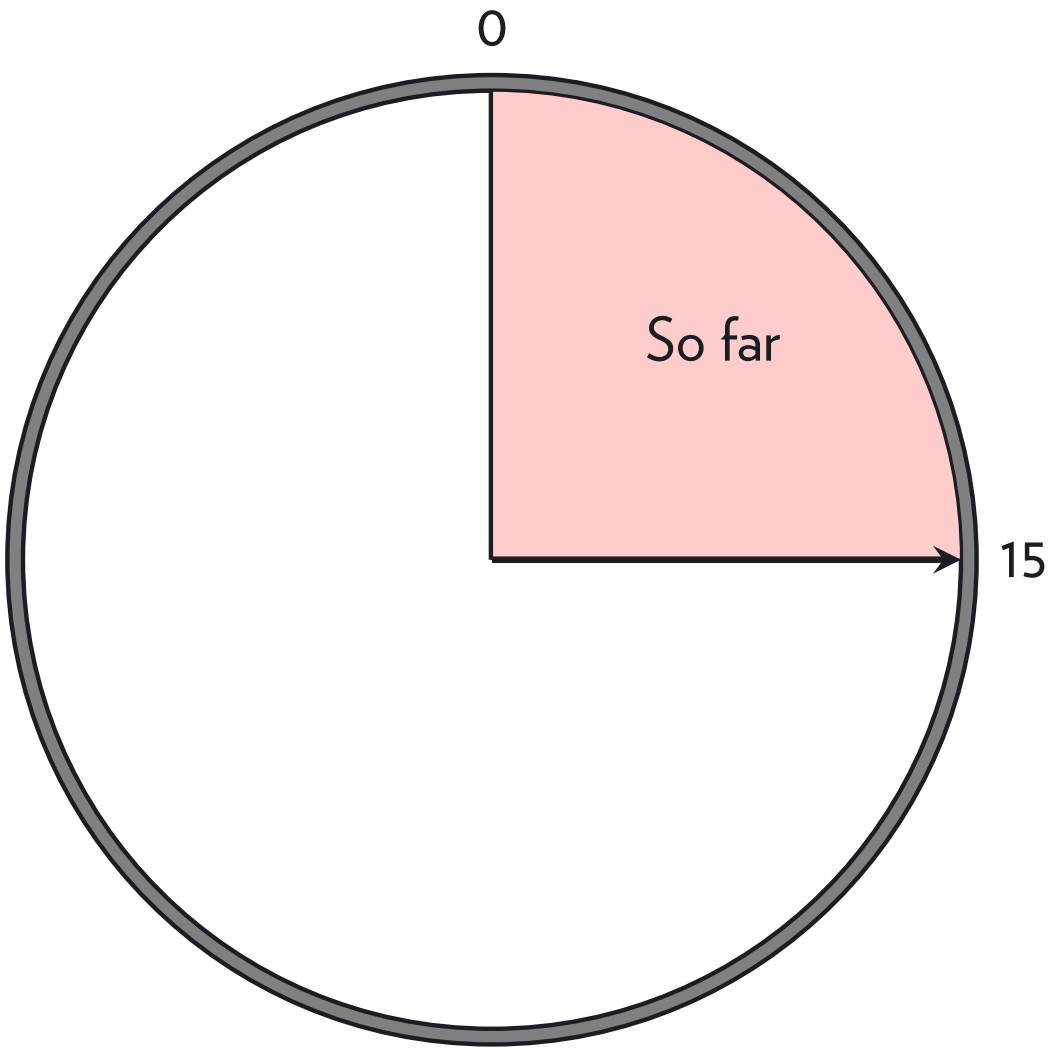
How well does Reed-Muller codes perform in the Shannon Model?

In BEC(p)?

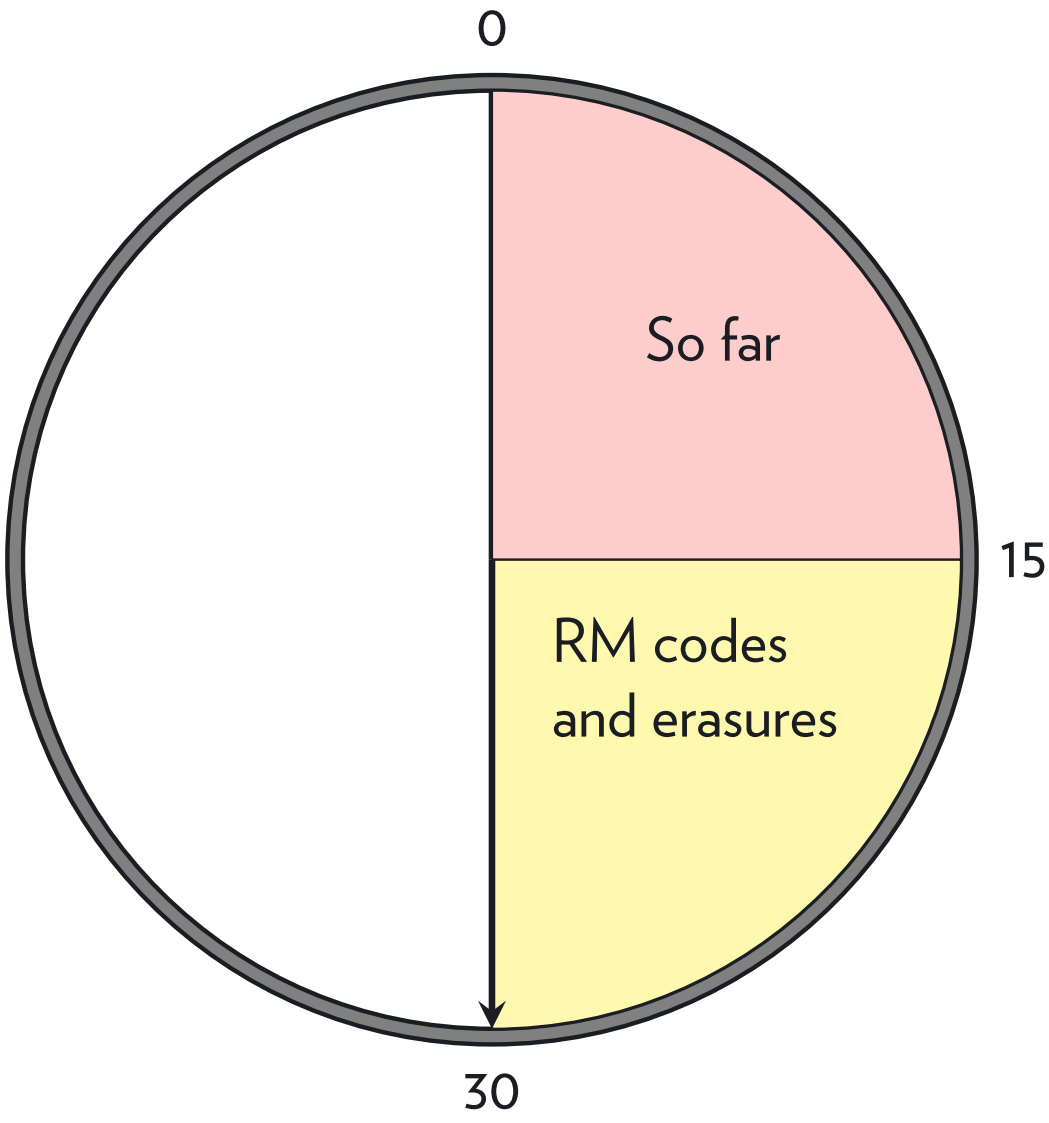
In BSC(p)?

Are they as good as polar codes?

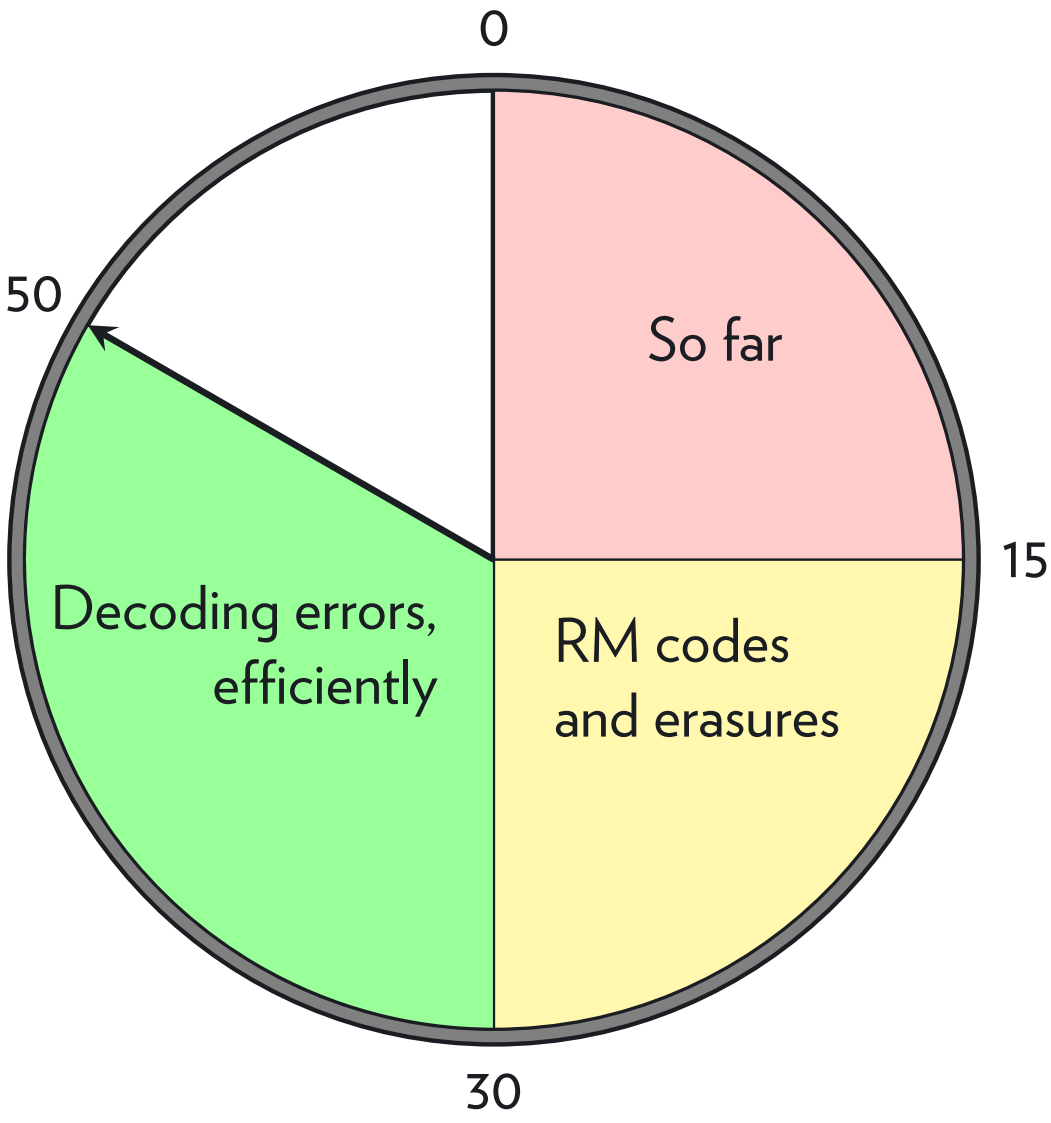
Outline



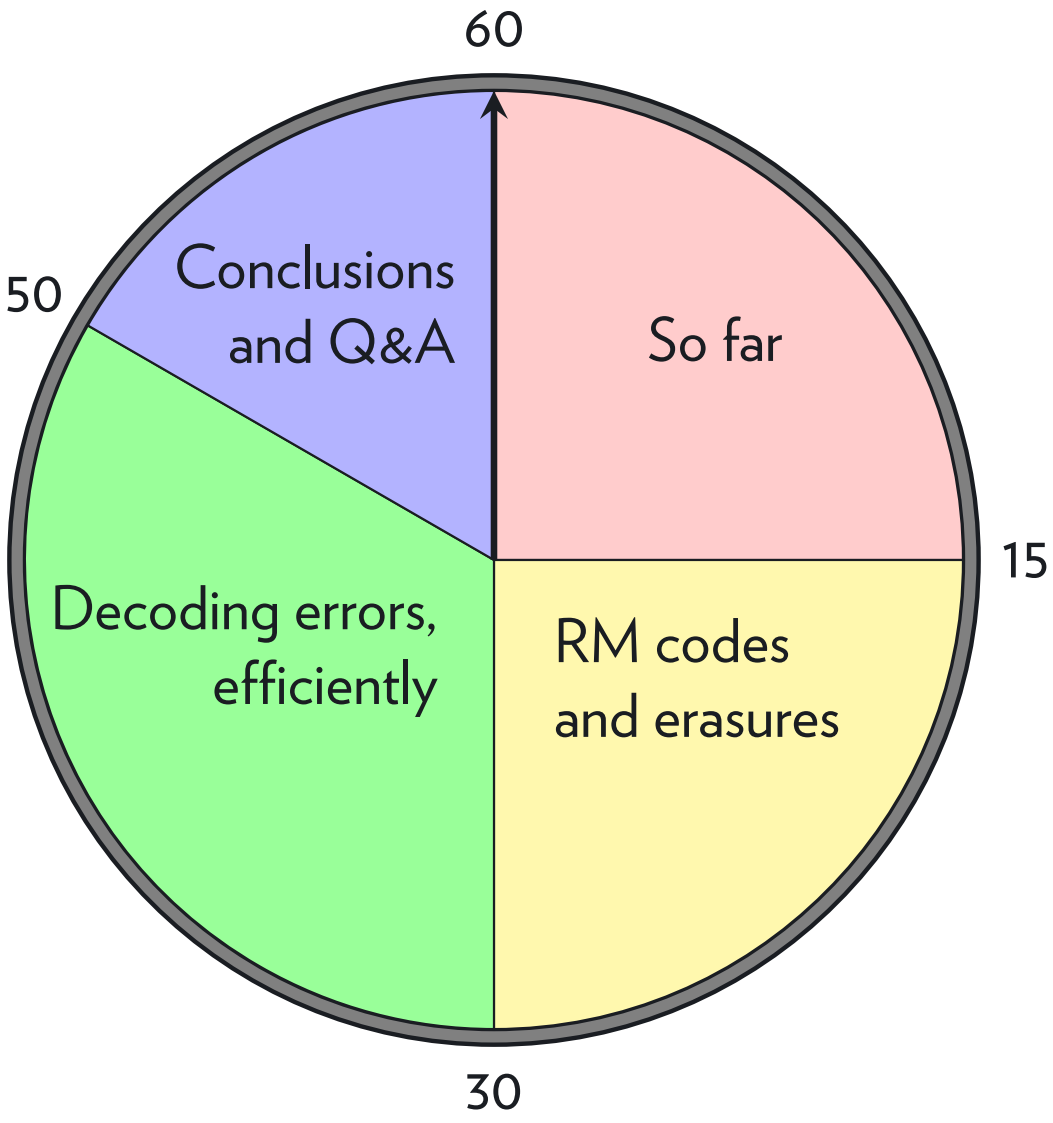
Outline



Outline


0
15
30
50
So far
RM codes
and erasures
Decoding errors,
efficiently

Outline


60
50
15
30
Conclusions and Q&A
So far
Decoding errors, efficiently
RM codes and erasures

Outline

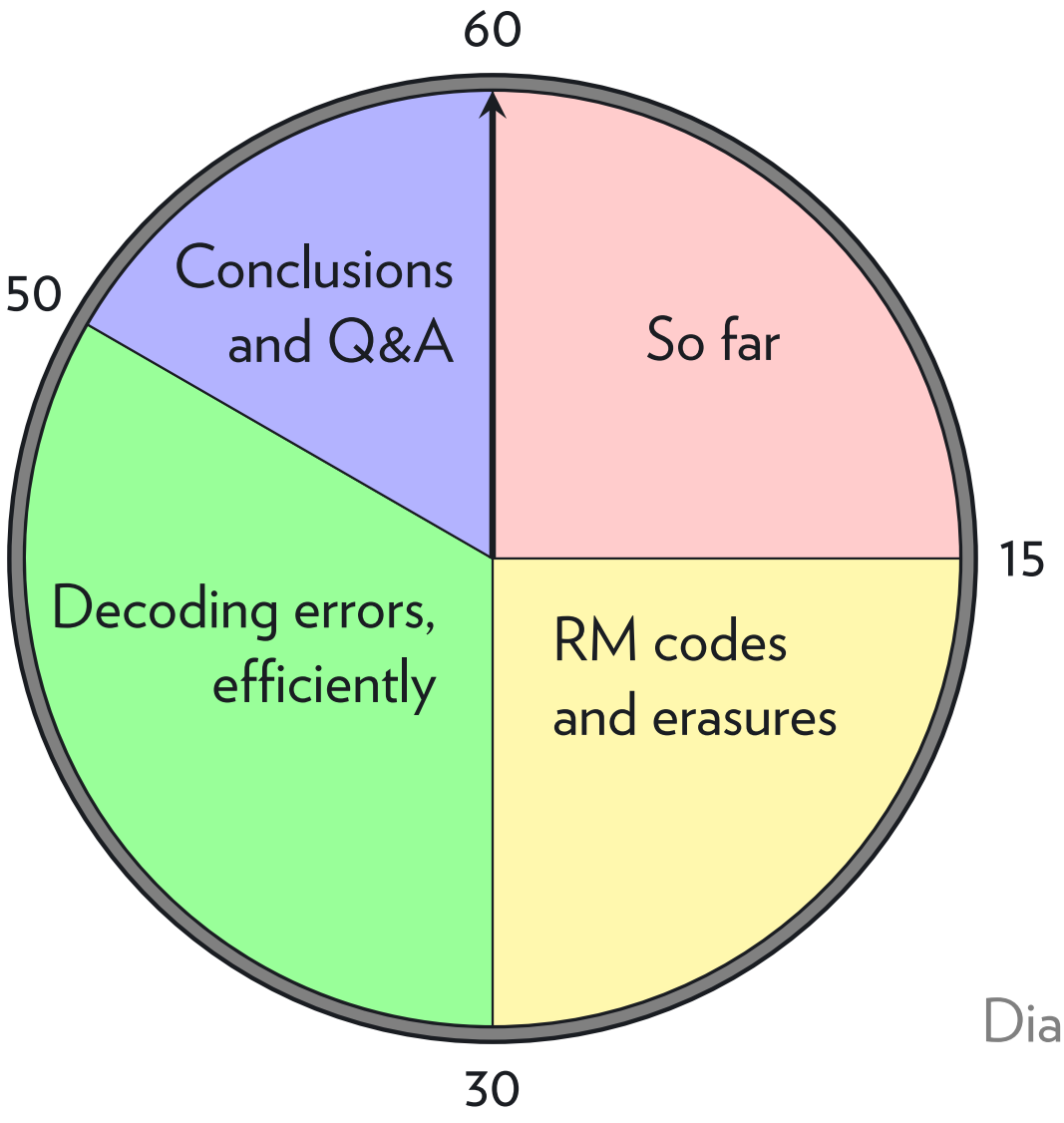


Diagram not to scale

Dual of a code

Dual of a code

Any linear space can be specified by a generating basis, or a solution to a system of constraints.

Dual of a code

Any linear space can be specified by a generating basis, or a solution to a system of constraints.

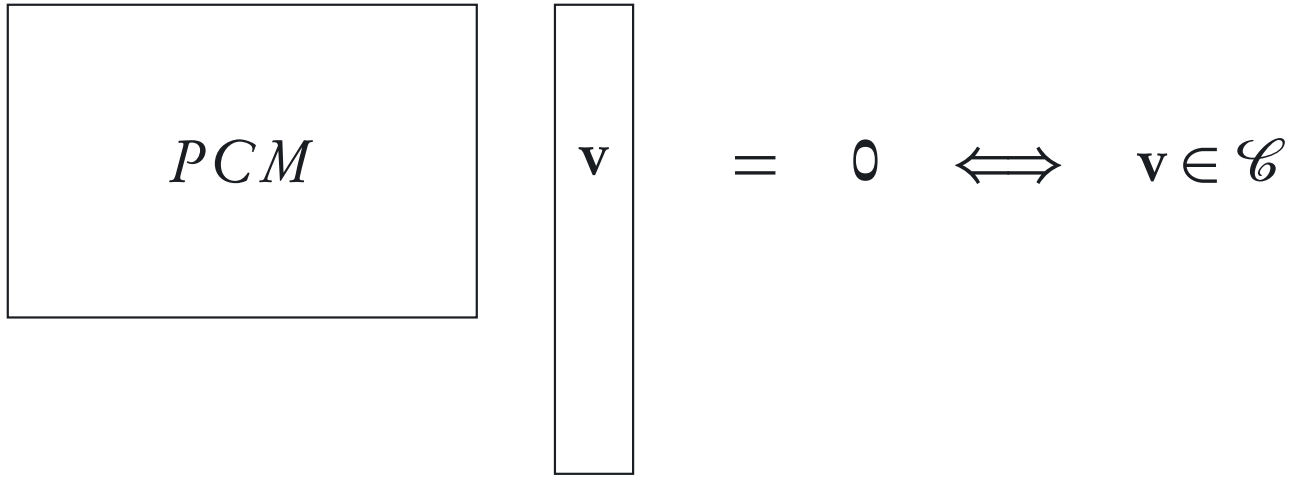
$$\mathscr{C}^{\perp} \quad = \quad \{\mathbf{u} \,:\, \langle \mathbf{v}, \mathbf{u} \rangle = 0 \text{ for every } v \in \mathscr{C}\}$$

Dual of a code

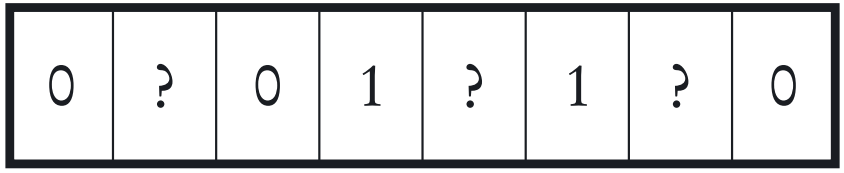
Any linear space can be specified by a generating basis, or a solution to a system of constraints.

$$\mathscr{C}^{\perp} \quad = \quad \{\mathbf{u} \,:\, \langle \mathbf{v}, \mathbf{u} \rangle = 0 \text{ for every } v \in \mathscr{C}\}$$

Parity Check Matrix
(A basis for $\mathscr{C}^{\perp}$ stacked as rows)



Linear codes and erasures



Question: When can we decode from a pattern of erasures?

Linear codes and erasures

0	?	0	1	?	1	?	0

Question: When can we decode from a pattern of erasures?

0	0	0	1	1	1	0	0

0	1	0	1	0	1	0	0

Linear codes and erasures

0	?	0	1	?	1	?	0

Question: When can we decode from a pattern of erasures?

0	0	0	1	1	1	0	0

0	1	0	1	0	1	0	0

0	1	0	0	1	0	0	0

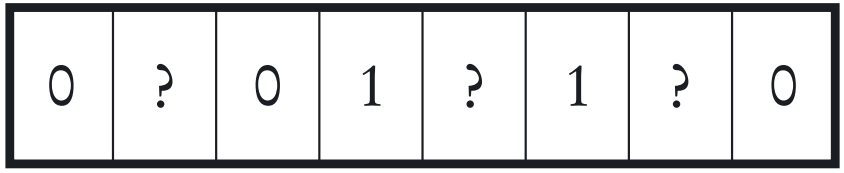
Linear codes and erasures

0	?	0	1	?	1	?	0

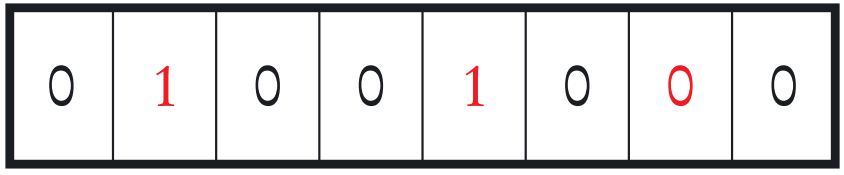
Question: When can we decode from a pattern of erasures?

0	1	0	0	1	0	0	0

Linear codes and erasures

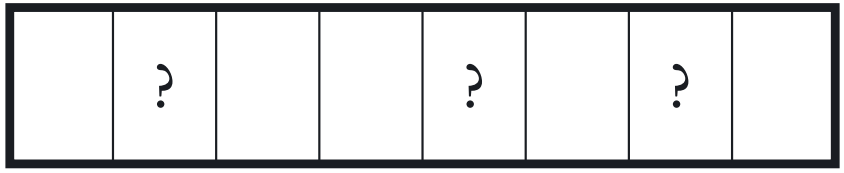


Question: When can we decode from a pattern of erasures?

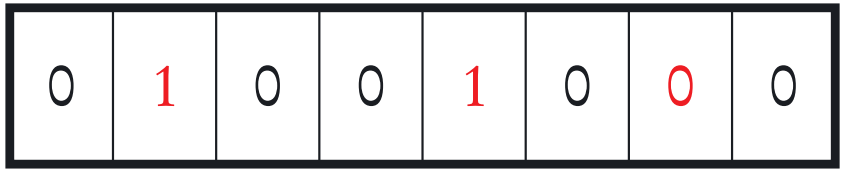


Decodable *if and only if* no non-zero codeword supported on erasures.

Linear codes and erasures



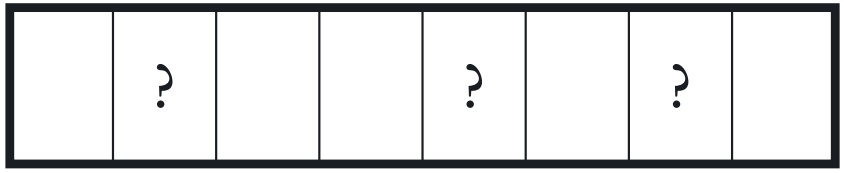
Question: When can we decode from a pattern of erasures?



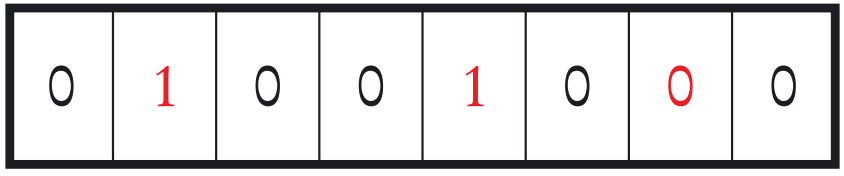
Decodable *if and only if* no non-zero codeword supported on erasures.

Depends only the erasure pattern

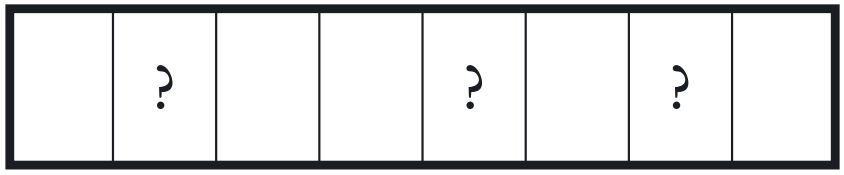
Linear codes and erasures



Question: When can we decode from a pattern of erasures?



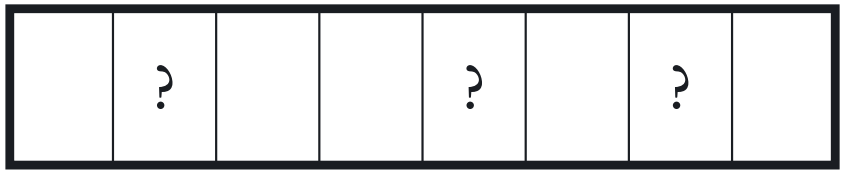
Linear codes and erasures



Question: When can we decode from a pattern of erasures?

Observation: A pattern of erasures are decodeable *if and only if* the corresponding columns of the Parity Check Matrix are linearly independent.

Linear codes and erasures



Question: When can we decode from a pattern of erasures?

Observation: A pattern of erasures are decodeable *if and only if* the corresponding columns of the Parity Check Matrix are linearly independent.

> *In order for a code to be good for* BEC(p), *the Parity Check Matrix of the code must be "robustly high-rank".*

Reed-Muller codes under erasures

Cool Fact

The dual of $RM(m, r)$ is $RM(m, r')$ where $r' = m - r - 1$.

Reed-Muller codes under erasures

Cool Fact

The dual of $RM(m, r)$ is $RM(m, r')$ where $r' = m - r - 1$.

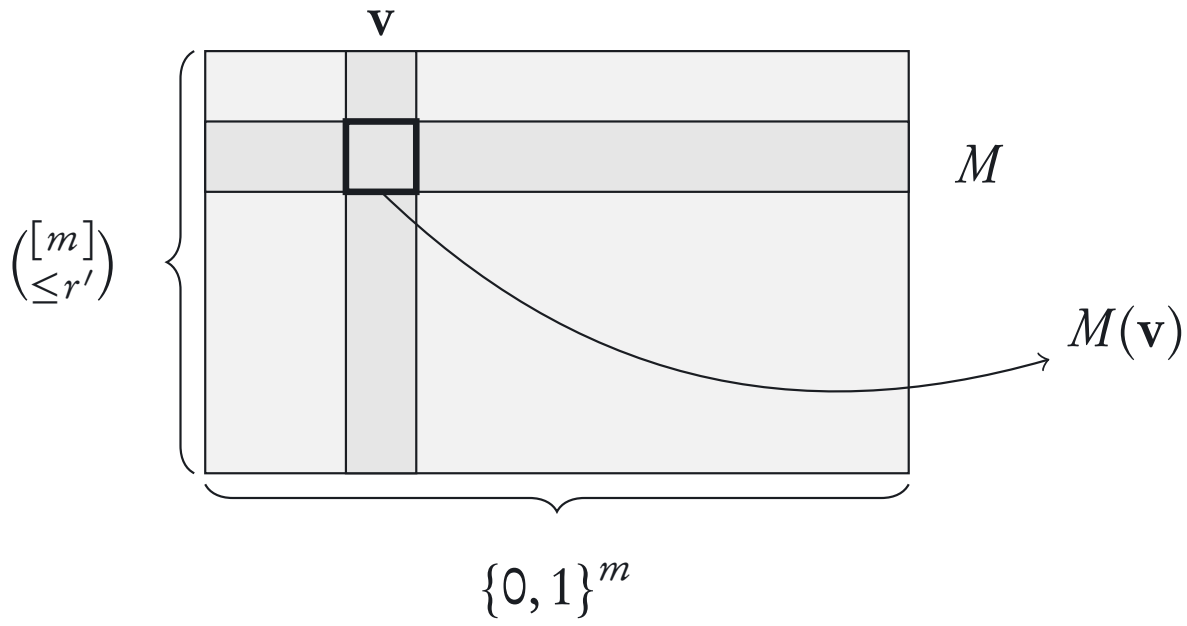
Hence, the Parity Check Matrix of $RM(m, r)$ is the generator matrix of $RM(m, r')$.

Reed-Muller codes under erasures

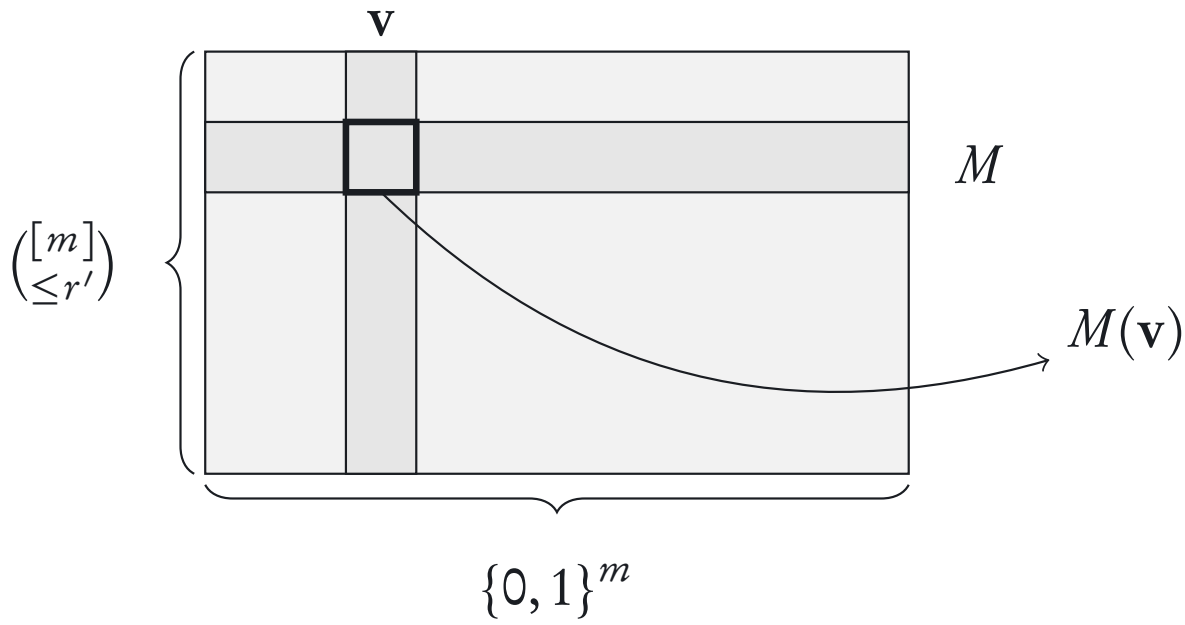
Cool Fact

The dual of $RM(m, r)$ is $RM(m, r')$ where $r' = m - r - 1$.

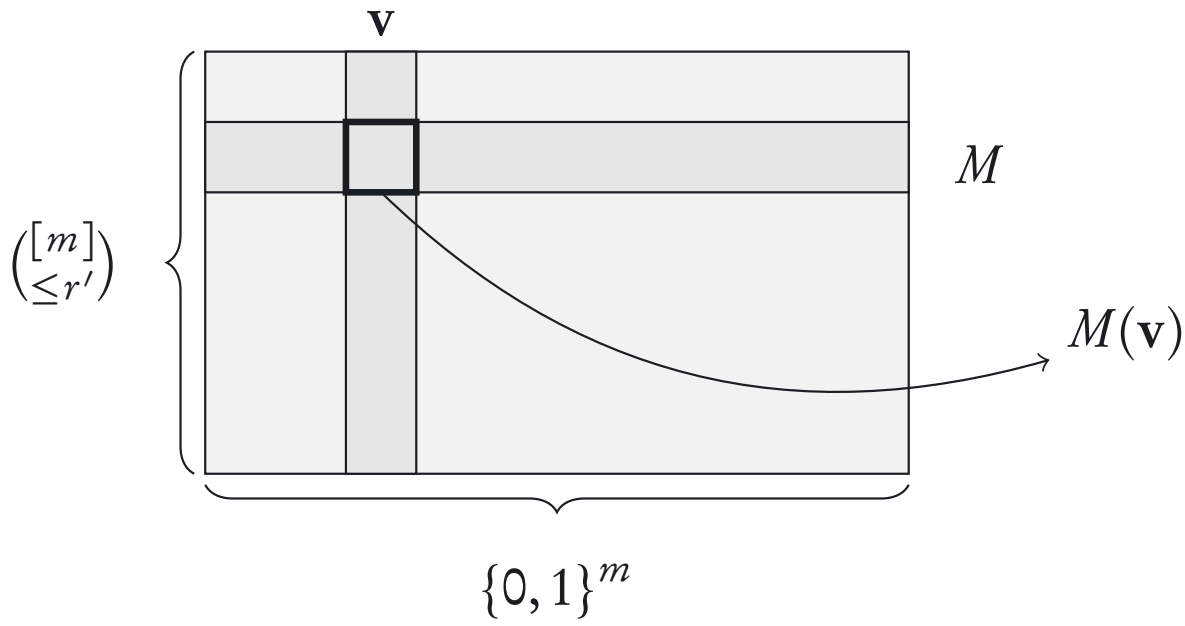
Hence, the Parity Check Matrix of $RM(m, r)$ is the generator matrix of $RM(m, r')$.



Reed-Muller codes under erasures

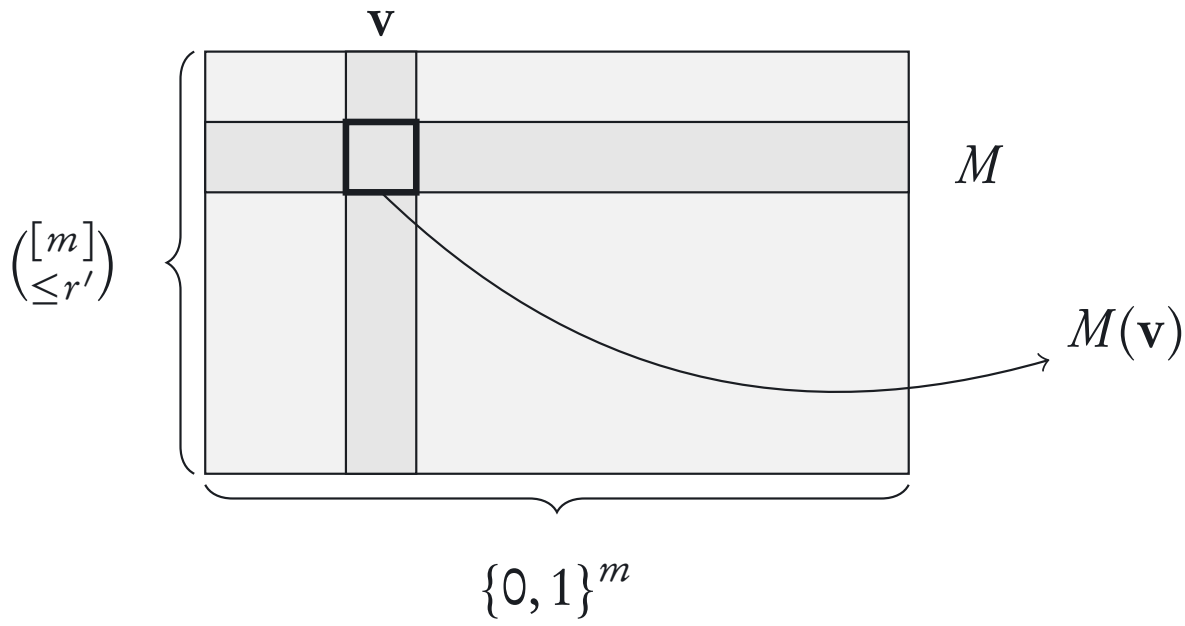


Reed-Muller codes under erasures

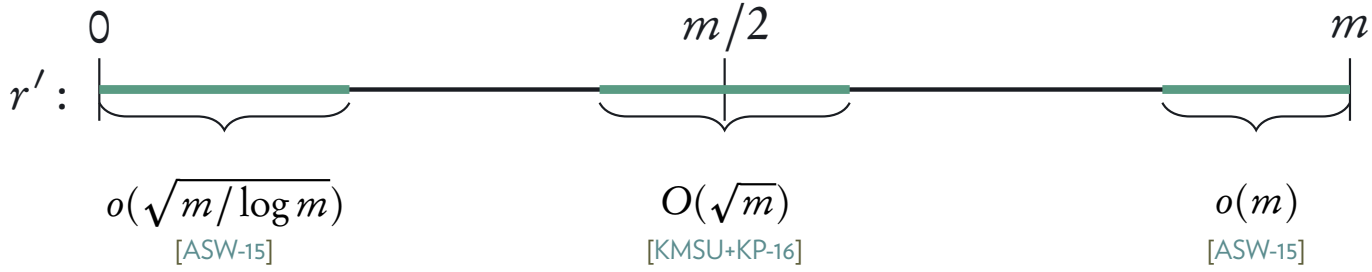


Question: Let $R = \binom{m}{\leq r'}$. Suppose you pick $(0.99)R$ columns at random. Are they linearly independent with high probability?

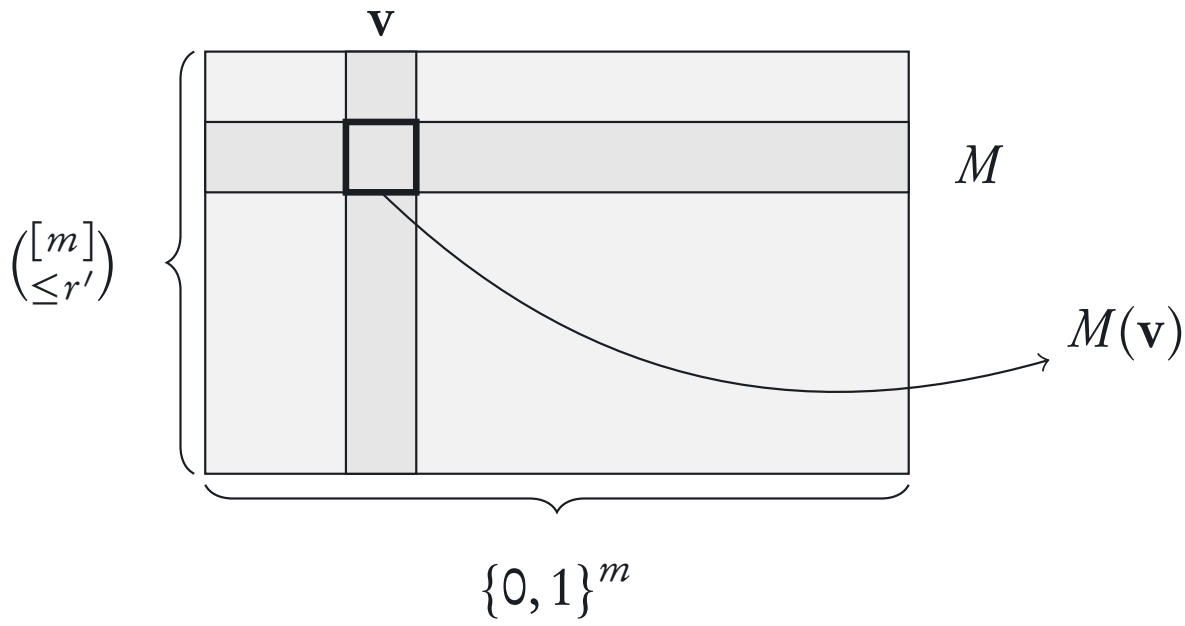
Reed-Muller codes under erasures



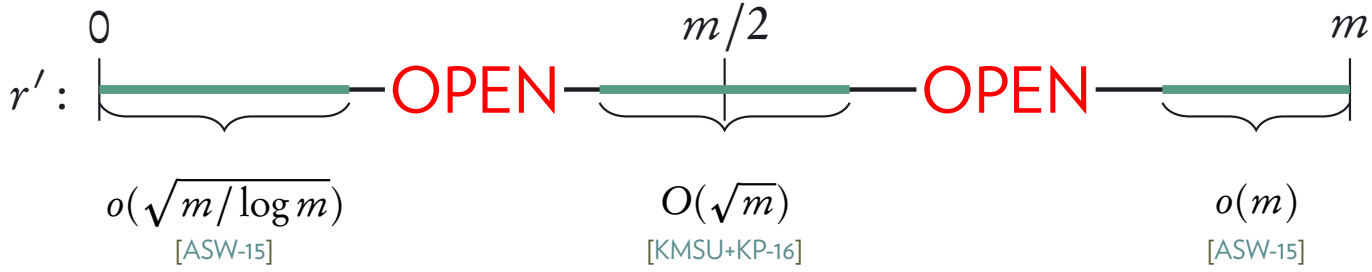
Question: Let $R = \binom{m}{\leq r'}$. Suppose you pick $(0.99)R$ columns at random. Are they linearly independent with high probability?



Reed-Muller codes under erasures



Question: Let $R = \binom{m}{\leq r'}$. Suppose you pick $(0.99)R$ columns at random. Are they linearly independent with high probability?

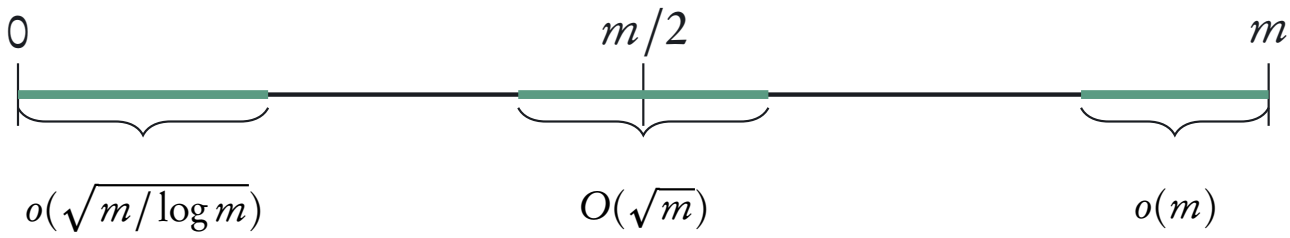


From erasures to errors

Theorem: [ASW] Any pattern correctable from erasures in $RM(m, m-r-1)$ is correctable from errors in $RM(m, m-2r-2)$.

From erasures to errors

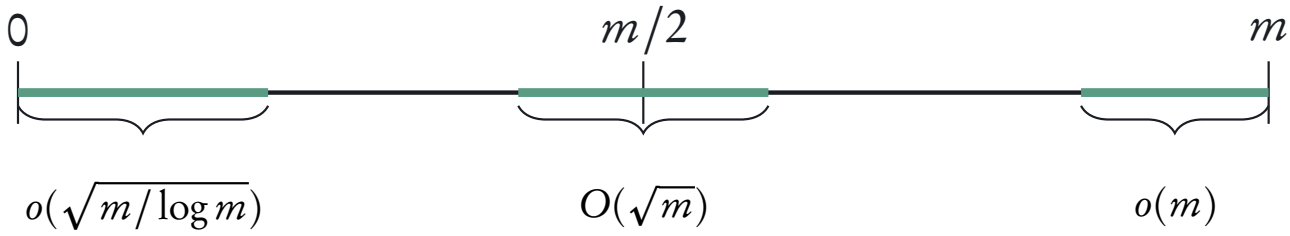
Remark: If r falls in the green zone, then $RM(m, m-r-1)$ can correct $\approx \binom{m}{\leq r}$ random erasures.



Theorem: [ASW] Any pattern correctable from erasures in $RM(m, m-r-1)$ is correctable from errors in $RM(m, m-2r-2)$.

From erasures to errors

Remark: If r falls in the green zone, then $RM(m, m-r-1)$ can correct $\approx \binom{m}{\leq r}$ random erasures.



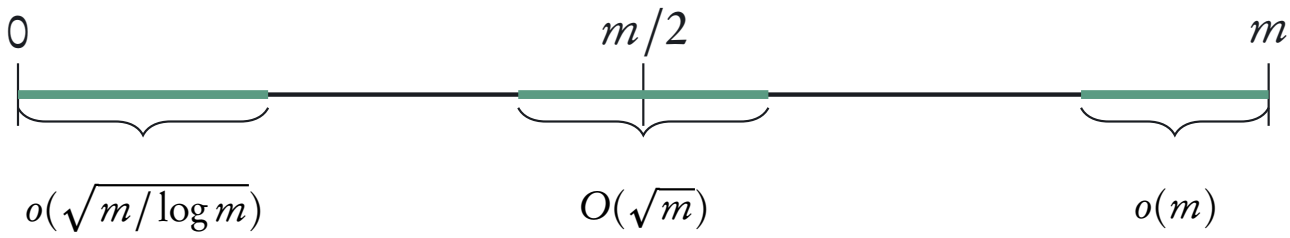
Theorem: [ASW] Any pattern correctable from erasures in $RM(m, m-r-1)$ is correctable from errors in $RM(m, m-2r-2)$.

Corollary #1: (high-rate) Decodeable from $(1-o(1))\binom{m}{\leq r}$ random errors in $RM(m, m-2r)$ if $r = o(\sqrt{m/\log m})$

(min distance of $RM(m, m-2r)$ is 2^{2r}) .

From erasures to errors

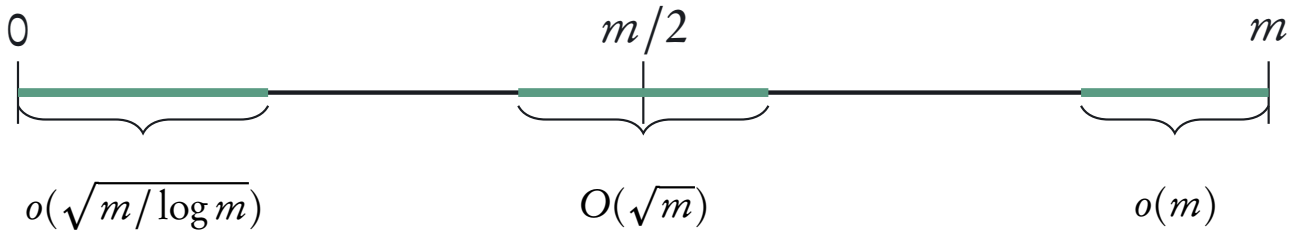
Remark: If r falls in the green zone, then $RM(m, m-r-1)$ can correct $\approx \binom{m}{\leq r}$ random erasures.



Theorem: [ASW] Any pattern correctable from erasures in $RM(m, m-r-1)$ is correctable from errors in $RM(m, m-2r-2)$.

From erasures to errors

Remark: If r falls in the green zone, then $RM(m, m-r-1)$ can correct $\approx \binom{m}{\leq r}$ random erasures.



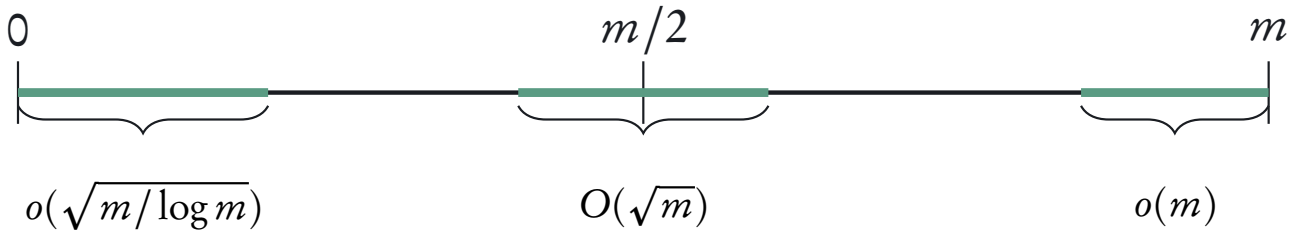
Theorem: [ASW] Any pattern correctable from erasures in $RM(m, m-r-1)$ is correctable from errors in $RM(m, m-2r-2)$.

(If $r = \frac{m}{2} - o(\sqrt{m})$, then $m-2r-2 = o(\sqrt{m})$)

Corollary #2: (low-rate) Decodeable from $(\frac{1}{2} - o(1))2^m$ random errors in $RM(m, o(\sqrt{m}))$ (min distance of $RM(m, \sqrt{m})$ is $2^{m-\sqrt{m}}$) .

From erasures to errors

Remark: If r falls in the green zone, then $RM(m, m-r-1)$ can correct $\approx \binom{m}{\leq r}$ random erasures.



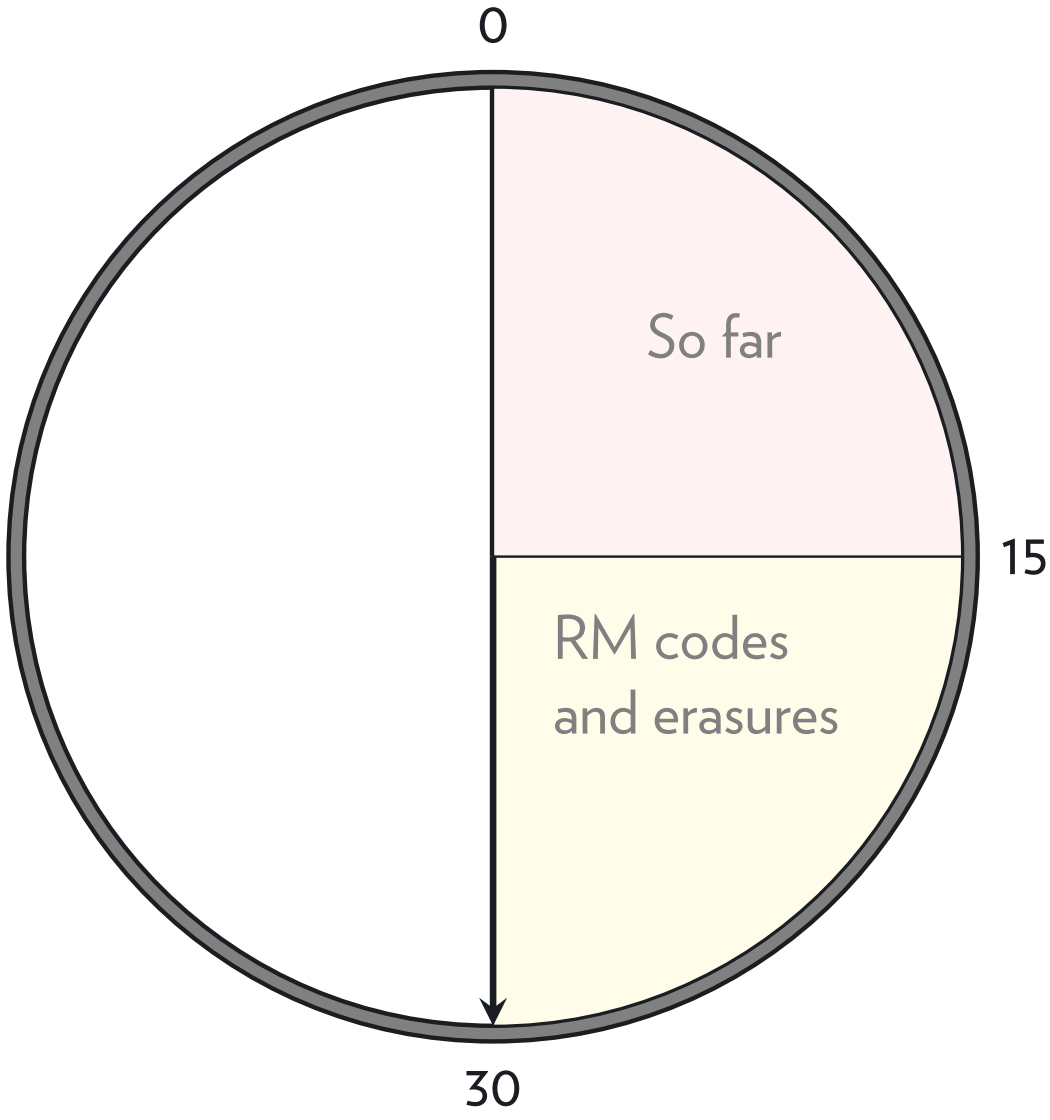
Theorem: [ASW] Any pattern correctable from erasures in $RM(m, m-r-1)$ is correctable from errors in $RM(m, m-2r-2)$.

(If $r = \frac{m}{2} - o(\sqrt{m})$, then $m - 2r - 2 = o(\sqrt{m})$)

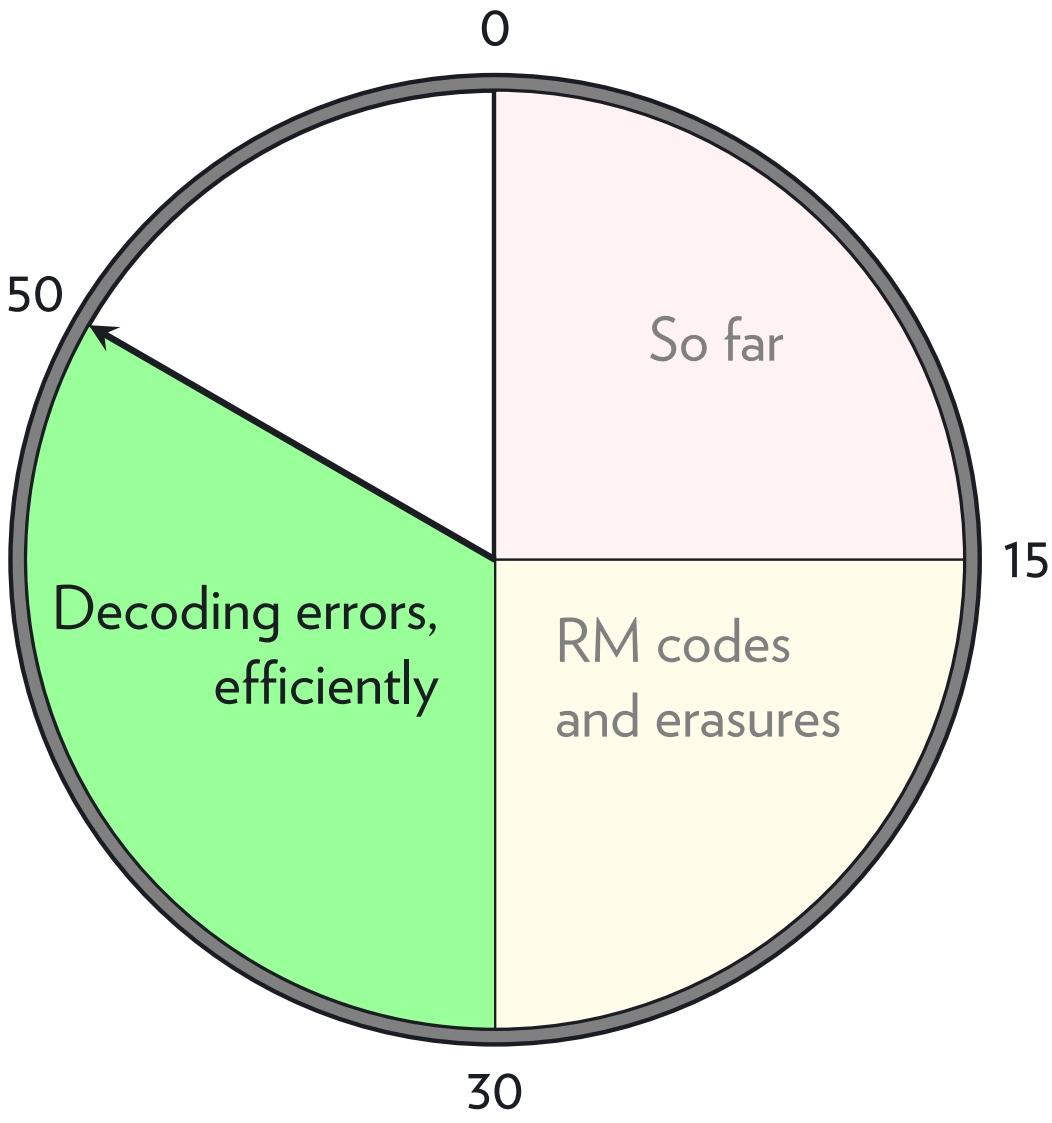
Corollary #2: (low-rate) Decodeable from $(\frac{1}{2} - o(1))2^m$ random errors in $RM(m, o(\sqrt{m}))$ (min distance of $RM(m, \sqrt{m})$ is $2^{m-\sqrt{m}}$) .

[S-Shpilka-Volk]: Efficient decoding from errors.

Outline



Outline


0
50
15
30
So far
Decoding errors,
efficiently
RM codes
and erasures

What we want to prove

Theorem [S-Shpilka-Volk]

There exists an efficient algorithm with the following guarantee:

> *Given a corrupted codeword* $\mathbf{w} = \mathbf{v} + \mathrm{err}_S$ *of* $RM(m, m-2r-1)$,
>
> *if* S *happens to be a correctable erasure pattern in* $RM(m, m-r-1)$,
>
> *then the algorithm correctly decodes* $\mathbf{v}$ *from* $\mathbf{w}$.

What we have access to

Received word is $\mathbf{w} := \mathbf{v} + \mathrm{err}_S$ for some $\mathbf{v} \in RM(m, m-2r-2)$ and $S = \{u_1, \ldots, u_t\}$.

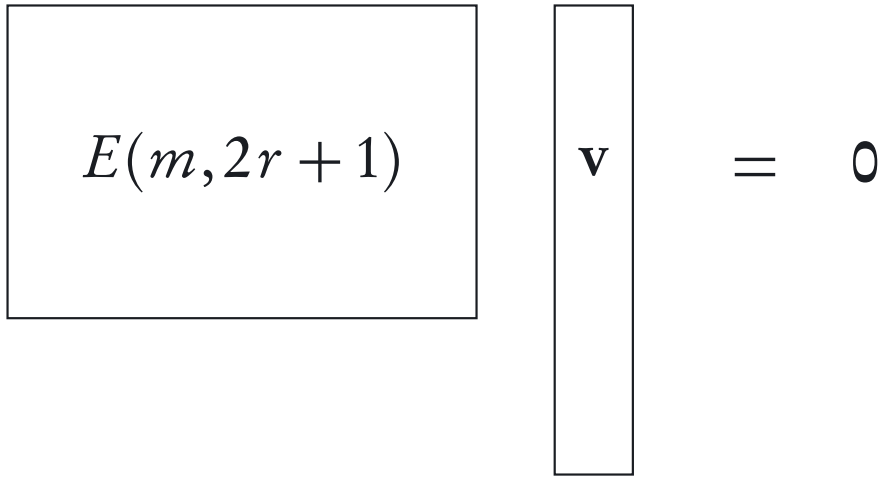
What we have access to

Received word is $\mathbf{w} := \mathbf{v} + \mathrm{err}_S$ for some $\mathbf{v} \in RM(m, m-2r-2)$ and $S = \{u_1, \ldots, u_t\}$.

Parity Check Matrix for $RM(m, m-2r-2)$ is the generator matrix for $RM(m, 2r+1)$.

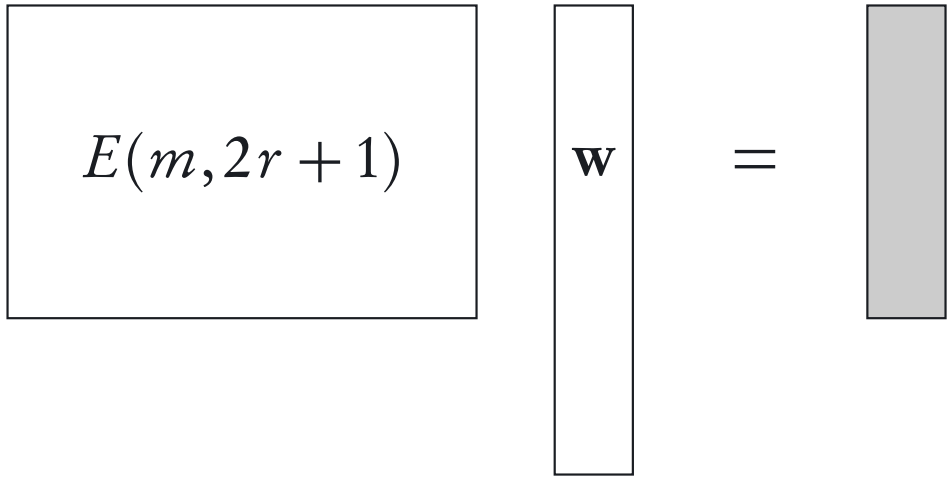
What we have access to

Received word is $\mathbf{w} := \mathbf{v} + \mathrm{err}_S$ for some $\mathbf{v} \in RM(m, m-2r-2)$ and $S = \{u_1, \ldots, u_t\}$.



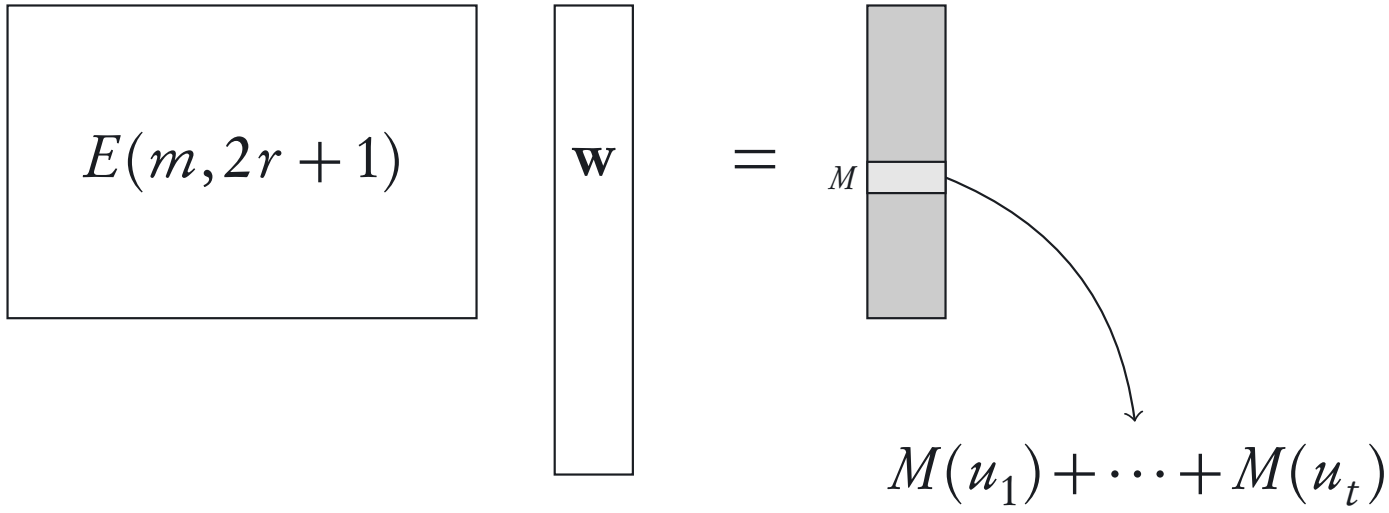
What we have access to

Received word is $\mathbf{w} := \mathbf{v} + \mathrm{err}_S$ for some $\mathbf{v} \in RM(m, m-2r-2)$ and $S = \{u_1, \ldots, u_t\}$.



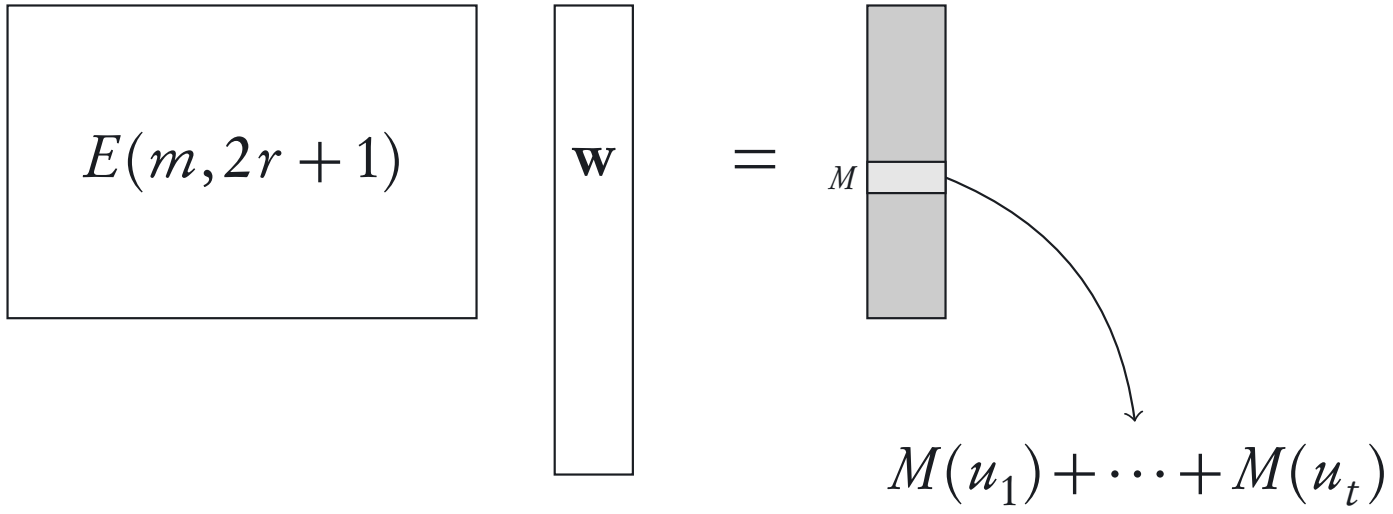
What we have access to

Received word is $\mathbf{w} := \mathbf{v} + \mathrm{err}_S$ for some $\mathbf{v} \in RM(m, m-2r-2)$ and $S = \{u_1, \ldots, u_t\}$.



What we have access to

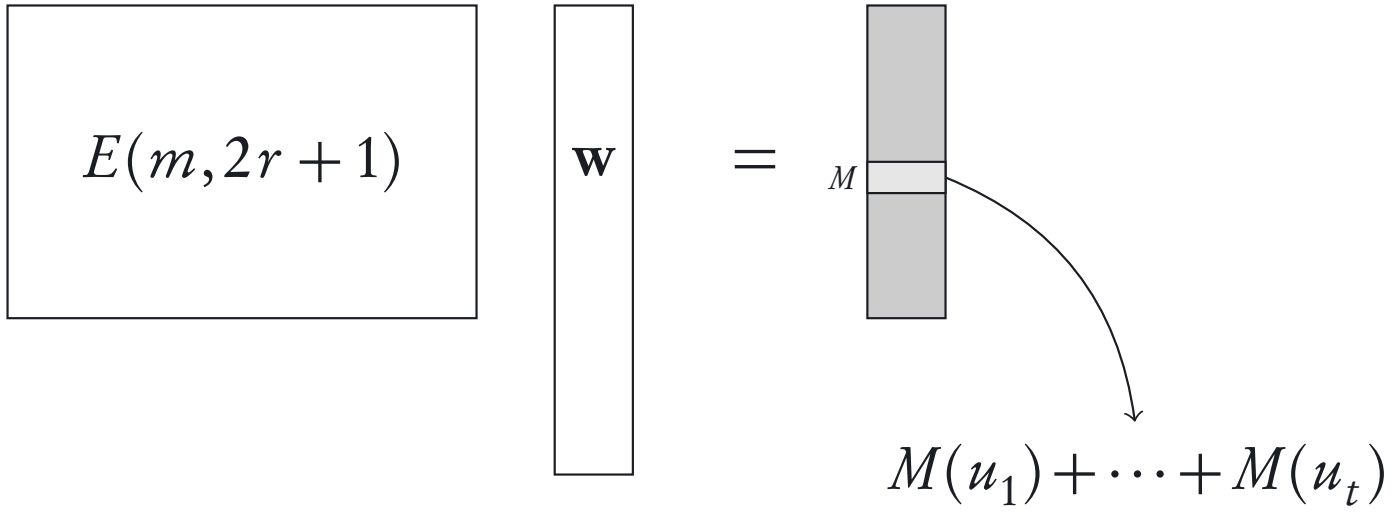
Received word is $\mathbf{w} := \mathbf{v} + \mathrm{err}_S$ for some $\mathbf{v} \in RM(m, m-2r-2)$ and $S = \{u_1, \ldots, u_t\}$.



Have access to $\sum_{i \in S} M(u_i)$ for every monomial M with $\deg(M) \leq 2r+1$.

What we have access to

Received word is $\mathbf{w} := \mathbf{v} + \mathrm{err}_S$ for some $\mathbf{v} \in RM(m, m-2r-2)$ and $S = \{u_1, \ldots, u_t\}$.



Have access to $\sum_{i \in S} f(u_i)$ for every polynomial f with $\deg(f) \leq 2r+1$.

Erasure Correctable Patterns

Erasure Correctable Patterns

A pattern of erasures is correctable *if and only if* the corresponding columns in the parity check matrix are linearly independent.

Erasure Correctable Patterns

A pattern of erasures is correctable *if and only if* the corresponding columns in the parity check matrix are linearly independent.

The parity check matrix for $RM(m, m - r - 1)$ is the generator matrix for $RM(m, r)$.

Erasure Correctable Patterns

A pattern of erasures is correctable *if and only if* the corresponding columns in the parity check matrix are linearly independent.

The parity check matrix for $RM(m, m-r-1)$ is the generator matrix for $RM(m, r)$.

Corollary

A set of patterns $S = \{u_1, \ldots, u_t\}$ is erasure-correctable in $RM(m, m-r-1)$ *if and only if* $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

u_i^r is just the vector of degree r monomials evaluated at u_i

The Decoding Algorithm

Input: Received word $\mathbf{w}$ $\quad (= \mathbf{v} + \mathrm{err}_S)$

The Decoding Algorithm

Input: Received word $\mathbf{w}$ $\quad (= \mathbf{v} + \mathrm{err}_S)$

Lemma

Assume that S is a pattern of erasures correctable in $RM(m, m-r-1)$.

The Decoding Algorithm

Input: Received word $\mathbf{w}$ $\quad (= \mathbf{v} + \mathrm{err}_S)$

Lemma

Assume that S is a pattern of erasures correctable in $RM(m, m-r-1)$.

For any arbitrary $u \in \{0,1\}^m$, we have $u \in S$ if and only if there exists a polynomial g with $\deg(g) \leq r$ such that

$$\sum_{i \in S} (f \cdot g)(u_i) \quad = \quad f(u) \quad \text{for every } f \text{ with } \deg(f) \leq r+1.$$

The Decoding Algorithm

Input: Received word $\mathbf{w}$ $\quad (= \mathbf{v} + \mathrm{err}_S)$

Lemma

Assume that S is a pattern of erasures correctable in $RM(m, m-r-1)$.

For any arbitrary $u \in \{0,1\}^m$, we have $u \in S$ if and only if there exists a polynomial g with $\deg(g) \leq r$ such that

$$\sum_{i \in S} (f \cdot g)(u_i) \quad = \quad f(u) \quad \text{for every } f \text{ with } \deg(f) \leq r+1.$$

Can be checked by solving a system of linear equations.
Algorithm is straightforward.

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\left\{u_1^r, \ldots, u_t^r\right\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Proof of Lemma

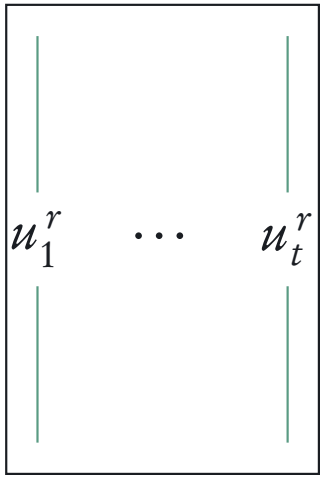
Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Proof.



□

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Proof.

□

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\left\{u_1^r, \ldots, u_t^r\right\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\left\{u_1^r, \ldots, u_t^r\right\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Main Lemma ($\Rightarrow$): If $u \in S$, then there is a polynomial g with $\deg(g) \leq r$ such that

$$\sum_{u_i \in S} (f \cdot g)(u_i) \quad = \quad f(u) \quad \textit{for every } f \textit{ with } \deg(f) \leq r+1.$$

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\left\{u_1^r, \ldots, u_t^r\right\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Main Lemma ($\Rightarrow$): If $u \in S$, then there is a polynomial g with $\deg(g) \leq r$ such that

$$\sum_{u_i \in S} (f \cdot g)(u_i) \quad = \quad f(u) \quad \text{for every } f \text{ with } \deg(f) \leq r+1.$$

If $u = u_i$, then $g = h_i$ satisfies the conditions.

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\left\{u_1^r, \ldots, u_t^r\right\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Claim 2

If $u \notin \{u_1, \ldots, u_t\}$, then there is a polynomial f such that $\deg(f) \leq r + 1$ and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \ldots, t.$$

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Claim 2

If $u \notin \{u_1, \ldots, u_t\}$, then there is a polynomial f such that $\deg(f) \leq r+1$ and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \ldots, t.$$

Main Lemma ($\Leftarrow$): If $u \notin S$, then there is **no** *polynomial g with $\deg(g) \leq r$ such that*

$$\sum_{u_i \in S} (f \cdot g)(u_i) \quad = \quad f(u) \quad \textit{for every } f \textit{ with } \deg(f) \leq r+1.$$

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Claim 2

If $u \notin \{u_1, \ldots, u_t\}$, then there is a polynomial f such that $\deg(f) \leq r + 1$ and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \ldots, t.$$

Proof

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

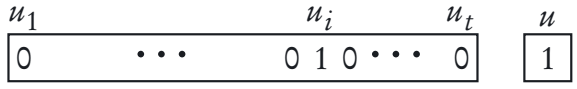
Claim 2

If $u \notin \{u_1, \ldots, u_t\}$, then there is a polynomial f such that $\deg(f) \leq r + 1$ and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \ldots, t.$$

Proof

Case 1: Suppose $h_i(u) = 1$ for some i.



Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

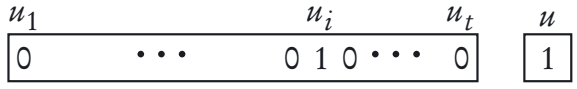
- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Claim 2

If $u \notin \{u_1, \ldots, u_t\}$, then there is a polynomial f such that $\deg(f) \leq r + 1$ and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \ldots, t.$$

Proof



Case 1: Suppose $h_i(u) = 1$ for some i.

$u \neq u_i$ hence $u_{(\ell)} \neq (u_i)_{(\ell)}$ for some coordinate ℓ.

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

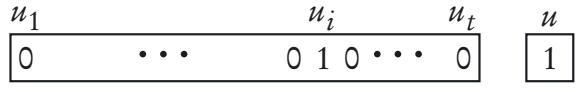
- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Claim 2

If $u \notin \{u_1, \ldots, u_t\}$, then there is a polynomial f such that $\deg(f) \leq r+1$ and

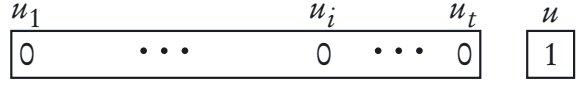
$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \ldots, t.$$

Proof



Case 1: Suppose $h_i(u) = 1$ for some i.

$u \neq u_i$ hence $u_{(\ell)} \neq (u_i)_{(\ell)}$ for some coordinate ℓ.



$f(\mathbf{x}) \quad = \quad h_i(\mathbf{x}) \cdot \left(x_\ell - (u_i)_{(\ell)}\right)$ works.

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Claim 2

If $u \notin \{u_1, \ldots, u_t\}$, then there is a polynomial f such that $\deg(f) \leq r + 1$ and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \ldots, t.$$

Proof

Case 2: Suppose $h_i(u) = 0$ for all i.

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Claim 2

If $u \notin \{u_1, \ldots, u_t\}$, then there is a polynomial f such that $\deg(f) \leq r + 1$ and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \ldots, t.$$

Proof

Case 2: Suppose $h_i(u) = 0$ for all i.

Look at $\sum h_i(\mathbf{x})$.

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Claim 2

If $u \notin \{u_1, \ldots, u_t\}$, then there is a polynomial f such that $\deg(f) \leq r+1$ and

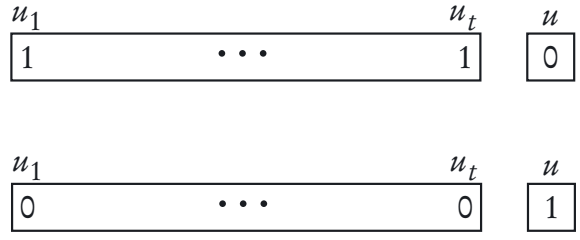
$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \ldots, t.$$

Proof

Case 2: Suppose $h_i(u) = 0$ for all i.

Look at $\sum h_i(\mathbf{x})$.

$f(\mathbf{x}) \quad = \quad 1 - \sum h_i(\mathbf{x})$ works.



Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Claim 2

If $u \notin \{u_1, \ldots, u_t\}$, then there is a polynomial f such that $\deg(f) \leq r+1$ and

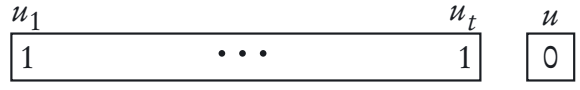
$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \ldots, t.$$

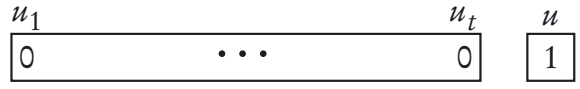
Proof

Case 2: Suppose $h_i(u) = 0$ for all i.

Look at $\sum h_i(\mathbf{x})$.

$f(\mathbf{x}) \quad = \quad 1 - \sum h_i(\mathbf{x})$ works.





□

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Claim 2

If $u \notin \{u_1, \ldots, u_t\}$, then there is a polynomial f such that $\deg(f) \leq r + 1$ and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \ldots, t.$$

Proof of Lemma

Claim 1

Let $u_1, \ldots, u_t \in \{0,1\}^m$ such that $\{u_1^r, \ldots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- $\deg(h_i) \leq r$,
- $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Claim 2

If $u \notin \{u_1, \ldots, u_t\}$, then there is a polynomial f such that $\deg(f) \leq r+1$ and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \ldots, t.$$

Therefore, if $\{u_1^r, \ldots, u_t^r\}$ are linearly independent, then there exists a polynomial g with $\deg(g) \leq r$ satisfying

$$\sum_{u_i \in S} (f \cdot g)(u_i) \quad = \quad f(u), \text{ for every polynomial } f \text{ with } \deg(f) \leq r+1,$$

if and only if $u = u_i$ for some i. □

Decoding Algorithm

- **Input:** Received word $\mathbf{w}$ $\quad (= \mathbf{v} + \mathrm{err}_S)$

Decoding Algorithm

- **Input:** Received word $\mathbf{w}$ $\quad (= \mathbf{v} + \mathrm{err}_S)$

- Compute $E(m, 2r+1)\mathbf{w}$ to get the value of $\sum_{u_i \in S} f(u_i)$ for every polynomial f with $\deg(f) \leq 2r+1$.

Decoding Algorithm

- **Input:** Received word $\mathbf{w}$ $\quad (= \mathbf{v} + \mathrm{err}_S)$

- Compute $E(m, 2r+1)\mathbf{w}$ to get the value of $\sum_{u_i \in S} f(u_i)$ for every polynomial f with $\deg(f) \leq 2r+1$.

- For each $u \in \{0,1\}^m$, solve for a polynomial g with $\deg(g) \leq r$ satisfying

$$\sum_{u_i \in S} (f \cdot g)(u_i) \quad = \quad f(u)\text{, for every } f \text{ satisfying } \deg(f) \leq r+1.$$

  If there is a solution, then add u to Corruptions.

Decoding Algorithm

- **Input:** Received word $\mathbf{w}$ $\quad (= \mathbf{v} + \mathrm{err}_S)$

- Compute $E(m, 2r+1)\mathbf{w}$ to get the value of $\sum_{u_i \in S} f(u_i)$ for every polynomial f with $\deg(f) \leq 2r+1$.

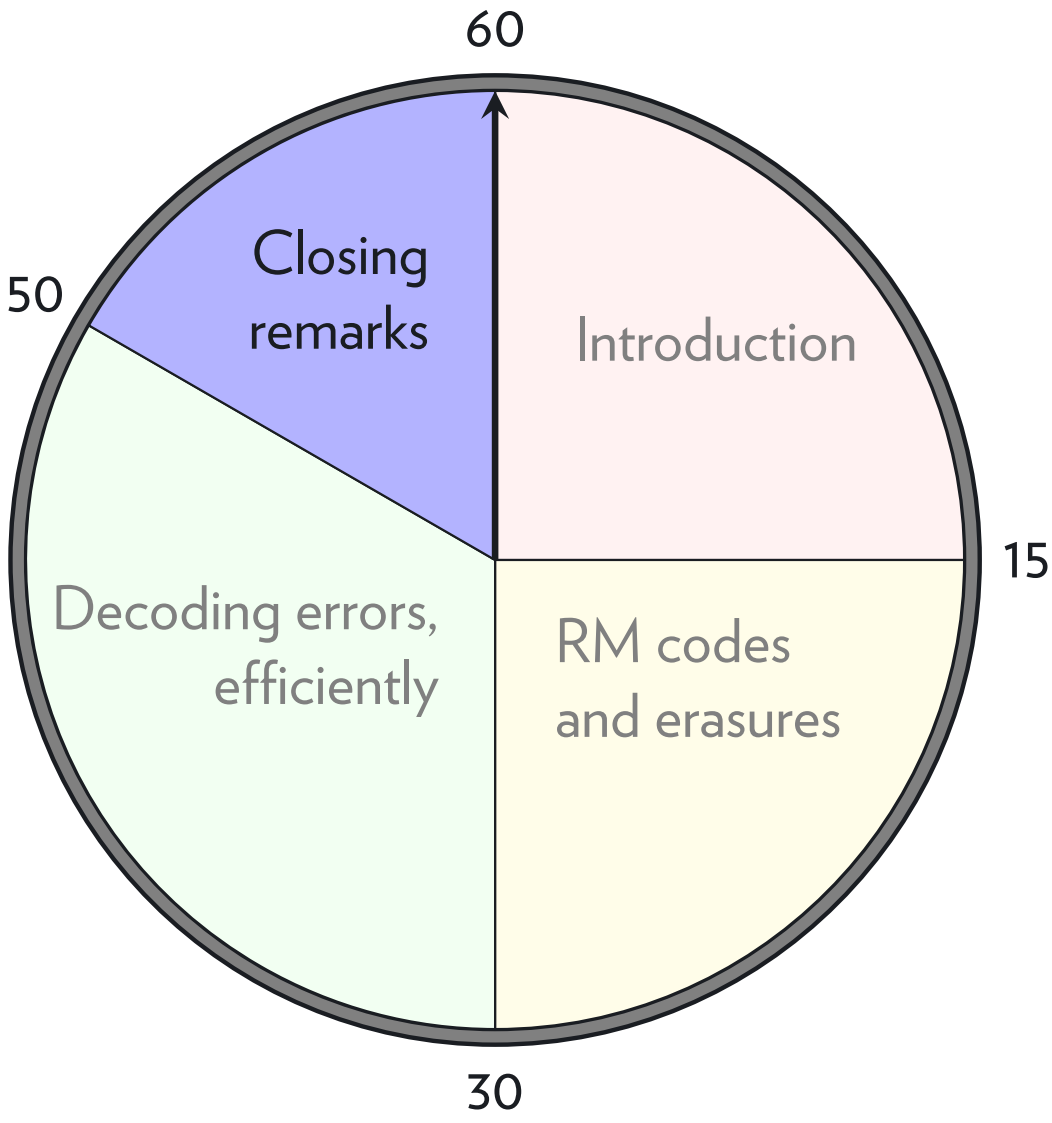
- For each $u \in \{0,1\}^m$, solve for a polynomial g with $\deg(g) \leq r$ satisfying

$$\sum_{u_i \in S} (f \cdot g)(u_i) \quad = \quad f(u)\text{, for every } f \text{ satisfying } \deg(f) \leq r+1.$$

  If there is a solution, then add u to Corruptions.

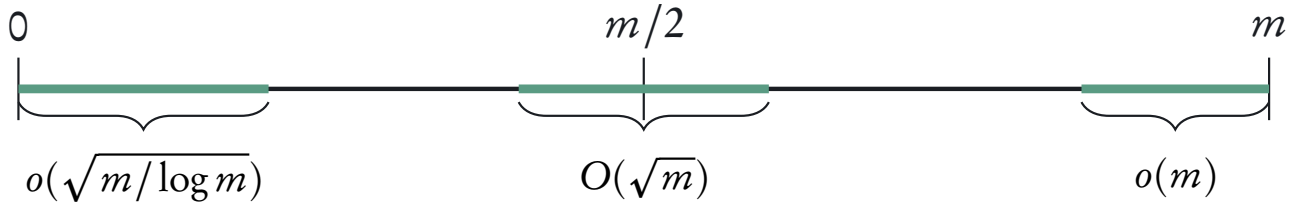
- Flip the coordinates in Corruptions and interpolate.

Outline


60
50
15
30
Closing remarks
Introduction
Decoding errors, efficiently
RM codes and erasures

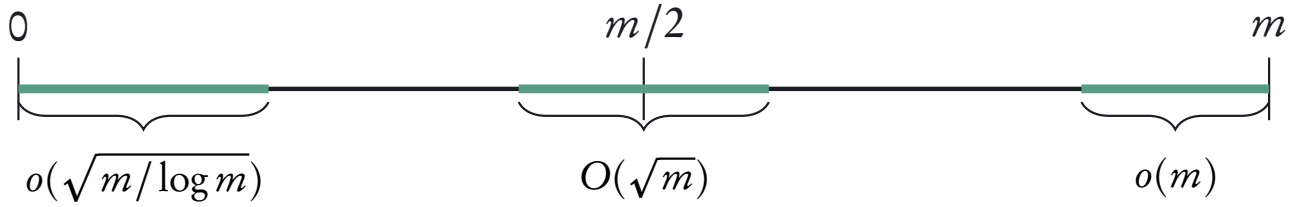
ICYMI

Remark: If r falls in the green zone, then $RM(m, m-r-1)$ can correct $\approx \binom{m}{\leq r}$ random erasures.



ICYMI

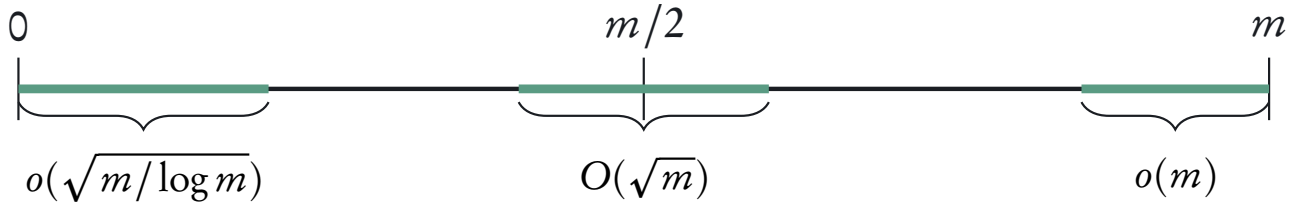
Remark: If r falls in the green zone, then $RM(m, m-r-1)$ can correct $\approx \binom{m}{\leq r}$ random erasures.



Theorem: [S-Shpilka-Volk] Any pattern that is erasure-correctable in $RM(m, m-r-1)$ is *efficiently* error-correctable in $RM(m, m-2r-2)$.

ICYMI

Remark: If r falls in the green zone, then $RM(m, m-r-1)$ can correct $\approx \binom{m}{\leq r}$ random erasures.



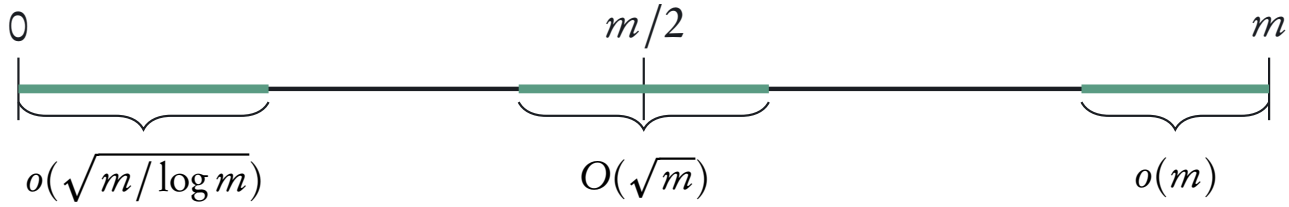
Theorem: [S-Shpilka-Volk] Any pattern that is erasure-correctable in $RM(m, m-r-1)$ is *efficiently* error-correctable in $RM(m, m-2r-2)$.

Corollary #1: (high-rate) Efficiently decodeable from $(1-o(1))\binom{m}{\leq r}$ random errors in $RM(m, m-2r)$ if $r = o(\sqrt{m/\log m})$.

(min distance of $RM(m, m-2r)$ is 2^{2r})

ICYMI

Remark: If r falls in the green zone, then $RM(m, m-r-1)$ can correct $\approx \binom{m}{\leq r}$ random erasures.



Theorem: [S-Shpilka-Volk] Any pattern that is erasure-correctable in $RM(m, m-r-1)$ is *efficiently* error-correctable in $RM(m, m-2r-2)$.

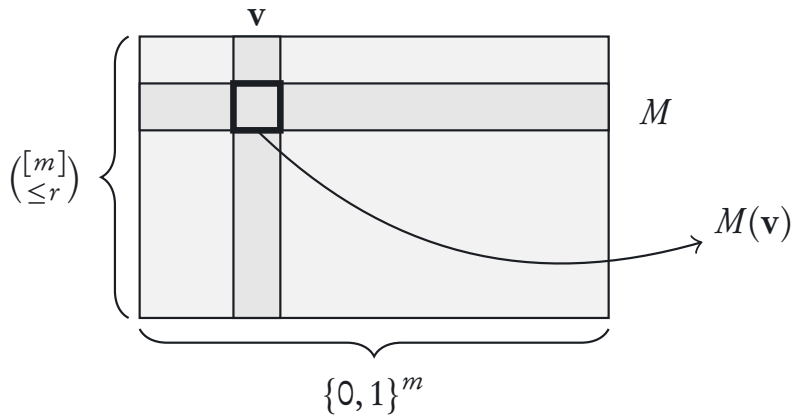
Corollary #1: (high-rate) Efficiently decodeable from $(1-o(1))\binom{m}{\leq r}$ random errors in $RM(m, m-2r)$ if $r = o(\sqrt{m/\log m})$.

(min distance of $RM(m, m-2r)$ is 2^{2r})

Corollary #2: (low-rate) Efficiently decodeable from $(\frac{1}{2} - o(1))2^m$ random errors in $RM(m, o(\sqrt{m}))$.

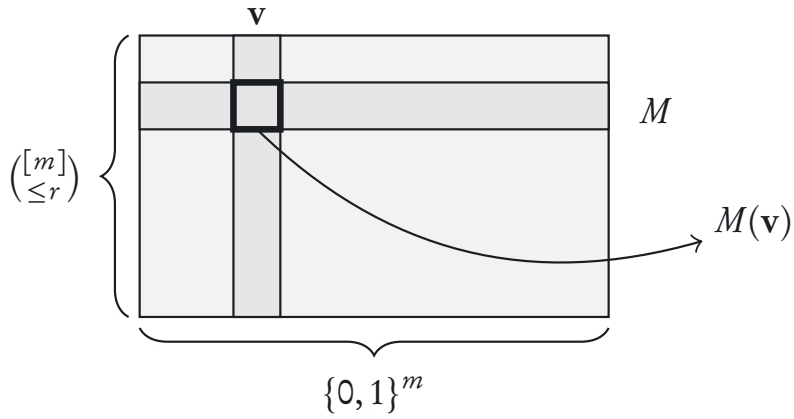
(min distance of $RM(m, \sqrt{m})$ is $2^{m-\sqrt{m}}$)

The obvious open question

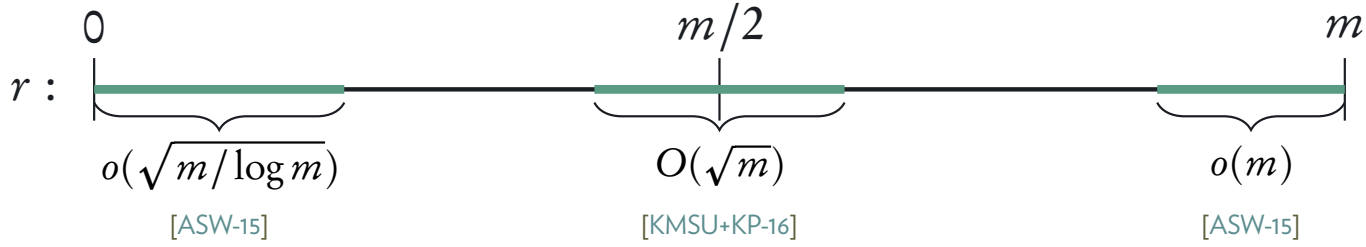


Question: Let $R = \binom{m}{\leq r}$. Suppose you pick $(0.99)R$ columns at random. Are they linearly independent with high probability?

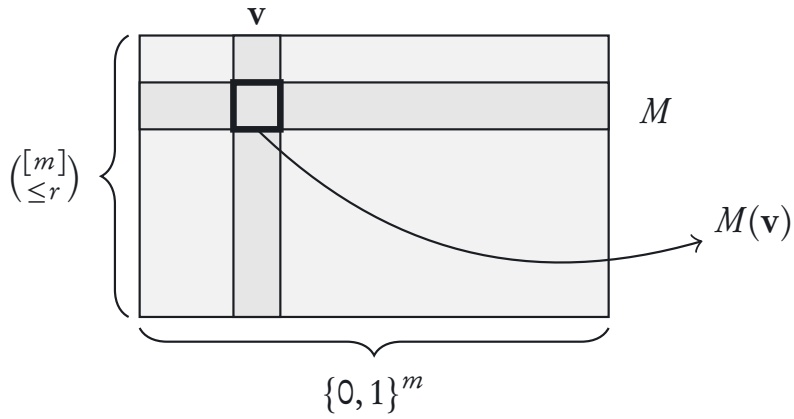
The obvious open question



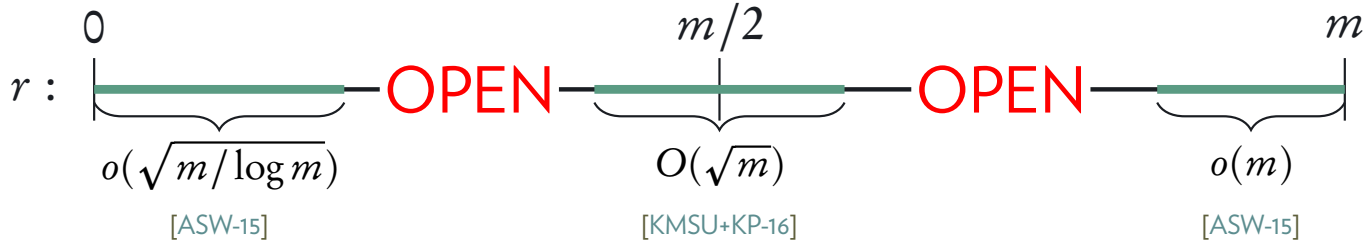
Question: Let $R = \binom{m}{\leq r}$. Suppose you pick $(0.99)R$ columns at random. Are they linearly independent with high probability?



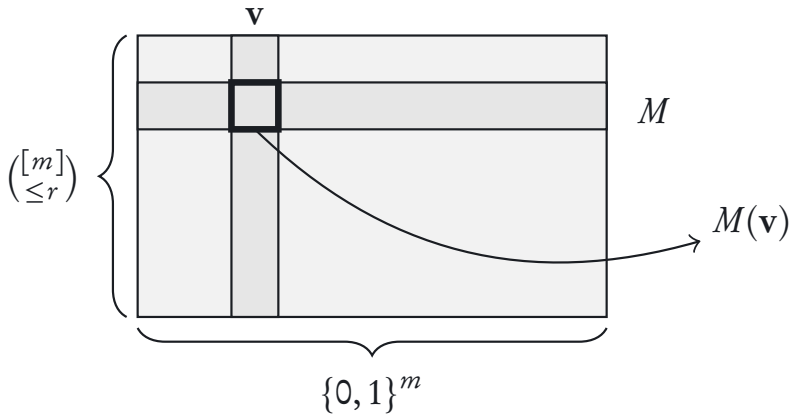
The obvious open question



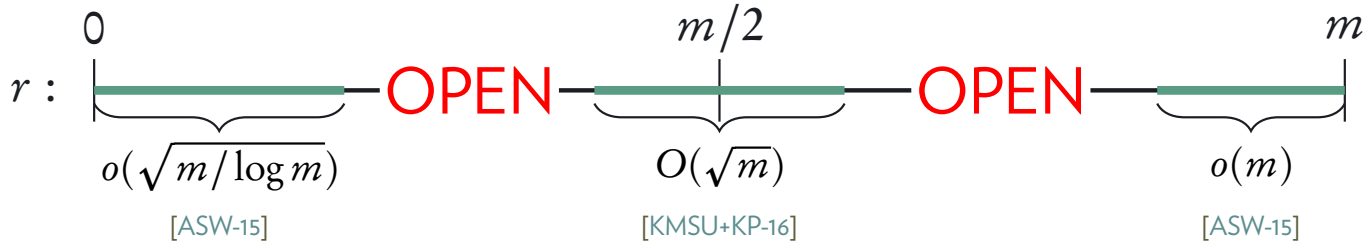
Question: Let $R = \binom{m}{\leq r}$. Suppose you pick $(0.99)R$ columns at random. Are they linearly independent with high probability?



The obvious open question



Question: Let $R = \binom{m}{\leq r}$. Suppose you pick $(0.99)R$ columns at random. Are they linearly independent with high probability?



\end{document}