EFFICIENTLY DECODING REED MULLER CODES FROM RANDOM ERRORS

Ben Lee Volk

Joint work with

Ramprasad Saptharishi
Amir Shpilka

Tel Aviv University

A GAME!

Given the truth-table of a polynomial $f \in \mathbb{F}[x_1, \ldots, x_m]$ of degree $\leq r$, with $1/2 - o(1)$ of the entries flipped, recover f **efficiently**.

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

A GAME!

Given the truth-table of a polynomial $f \in \mathbb{F}[x_1, \ldots, x_m]$ of degree $\leq r$, with $1/2 - o(1)$ of the entries flipped, recover f **efficiently**.

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

A GAME!

Given the truth-table of a polynomial $f \in \mathbb{F}[x_1, \ldots, x_m]$ of degree $\leq r$, with $1/2 - o(1)$ of the entries flipped, recover f **efficiently**.

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

A GAME!

Given the truth-table of a polynomial $f \in \mathbb{F}[x_1, \ldots, x_m]$ of degree $\leq r$, with $1/2 - o(1)$ of the entries flipped, recover f **efficiently**.

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

A GAME!

Given the truth-table of a polynomial $f \in \mathbb{F}[x_1, \ldots, x_m]$ of degree $\leq r$, with $1/2 - o(1)$ of the entries flipped, recover f **efficiently**.

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

Adversarial Errors: Impossible.

A GAME!

Given the truth-table of a polynomial $f \in \mathbb{F}[x_1, \ldots, x_m]$ of degree $\leq r$, with $1/2 - o(1)$ of the entries flipped, recover f **efficiently**.

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

Adversarial Errors: Impossible.

Random Errors?

(each coordinate flipped independently with probability $1/2 - o(1)$)

A GAME!

Given the truth-table of a polynomial $f \in \mathbb{F}[x_1, \ldots, x_m]$ of degree $\leq r$, with $1/2 - o(1)$ of the entries flipped, recover f **efficiently**.

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

Adversarial Errors: Impossible.
Random Errors?
(each coordinate flipped independently with probability $1/2 - o(1)$)

Theorem: There is an efficient algorithm to recover f, even for $r = o(\sqrt{m})$.

A GAME!

Given the truth-table of a polynomial $f \in \mathbb{F}[x_1, \ldots, x_m]$ of degree $\leq r$, with $1/2 - o(1)$ of the entries flipped, recover f **efficiently**.

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

Adversarial Errors: Impossible.
Random Errors?
(each coordinate flipped independently with probability $1/2 - o(1)$)

Theorem: There is an efficient algorithm to recover f, even for $r = o(\sqrt{m})$.

This talk is about decoding **Reed-Muller codes** from **random** errors.

REED-MULLER CODES: $RM(m, r)$

- **Messages:** (coefficient vectors of) degree $\leq r$ polynomials $f \in \mathbb{F}_2[x_1, \ldots, x_m]$

REED-MULLER CODES: $RM(m, r)$

- **Messages:** (coefficient vectors of) degree $\leq r$ polynomials $f \in \mathbb{F}_2[x_1, \ldots, x_m]$
- **Encoding:** evaluations over $\mathbb{F}_2^m$

REED-MULLER CODES: $RM(m, r)$

- **Messages:** (coefficient vectors of) degree $\leq r$ polynomials $f \in \mathbb{F}_2[x_1, \ldots, x_m]$
- **Encoding:** evaluations over $\mathbb{F}_2^m$

$$f(x_1, x_2, x_3, x_4) = x_1 x_2 + x_3 x_4$$
$$\downarrow$$

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

REED-MULLER CODES: $RM(m, r)$

- **Messages:** (coefficient vectors of) degree $\leq r$ polynomials $f \in \mathbb{F}_2[x_1, \ldots, x_m]$
- **Encoding:** evaluations over $\mathbb{F}_2^m$

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

$$\downarrow$$

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

A linear code, with

REED-MULLER CODES: $RM(m, r)$

- **Messages:** (coefficient vectors of) degree $\leq r$ polynomials $f \in \mathbb{F}_2[x_1, \ldots, x_m]$
- **Encoding:** evaluations over $\mathbb{F}_2^m$

$$f(x_1, x_2, x_3, x_4) = x_1 x_2 + x_3 x_4$$
$$\downarrow$$

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

A linear code, with

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

REED-MULLER CODES: $RM(m, r)$

- **Messages:** (coefficient vectors of) degree $\leq r$ polynomials $f \in \mathbb{F}_2[x_1, \ldots, x_m]$
- **Encoding:** evaluations over $\mathbb{F}_2^m$

$$f(x_1, x_2, x_3, x_4) = x_1 x_2 + x_3 x_4$$
$$\downarrow$$

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

A linear code, with

- **Block Length:** $2^m := n$
- **Distance:** 2^{m-r} (lightest codeword: $x_1 x_2 \cdots x_r$)

REED-MULLER CODES: $RM(m, r)$

- **Messages:** (coefficient vectors of) degree $\leq r$ polynomials $f \in \mathbb{F}_2[x_1, \ldots, x_m]$
- **Encoding:** evaluations over $\mathbb{F}_2^m$

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

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

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: $RM(m, r)$

- **Messages:** (coefficient vectors of) degree $\leq r$ polynomials $f \in \mathbb{F}_2[x_1, \ldots, x_m]$
- **Encoding:** evaluations over $\mathbb{F}_2^m$

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

$$\downarrow$$

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

A linear code, with

- **Block Length:** $2^m := n$
- **Distance:** 2^{m-r} (lightest codeword: $x_1 x_2 \cdots x_r$)
- **Dimension:** $\binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{r} := \binom{m}{\leq r}$
- **Rate:** dimension/block length $= \binom{m}{\leq r} / 2^m$

MORE PROPERTIES OF REED-MULLER CODES

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

$$\begin{array}{cc} & \mathbf{v} \in \mathbb{F}_2^m \\ M & \left(\begin{array}{ccc} & & \\ & M(\mathbf{v}) & \\ & & \end{array} \right) := E(m,r) \end{array}$$

MORE PROPERTIES OF REED-MULLER CODES

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

$$\begin{array}{cc} & \mathbf{v} \in \mathbb{F}_2^m \\ M & \begin{pmatrix} & & \\ & M(\mathbf{v}) & \\ & & \end{pmatrix} := E(m,r) \end{array}$$

(Every codeword is spanned by the rows.)

MORE PROPERTIES OF REED-MULLER CODES

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

$$\begin{array}{cc} & \mathbf{v} \in \mathbb{F}_2^m \\ M & \begin{pmatrix} & & \\ & M(\mathbf{v}) & \\ & & \end{pmatrix} \end{array} := E(m, r)$$

(Every codeword is spanned by the rows.)

Linear codes can be also defined by their **parity check matrix**: a matrix H whose kernel is the code.

MORE PROPERTIES OF REED-MULLER CODES

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

$$\begin{array}{cc} & \mathbf{v} \in \mathbb{F}_2^m \\ M & \begin{pmatrix} & & \\ & M(\mathbf{v}) & \\ & & \end{pmatrix} \end{array} := E(m, r)$$

(Every codeword is spanned by the rows.)

Linear codes can be also defined by their **parity check matrix**: a matrix H whose kernel is the code.

Duality: $RM(m, r)^{\perp} = RM(m, m - r - 1)$.

MORE PROPERTIES OF REED-MULLER CODES

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

$$\begin{array}{cc} & \mathbf{v} \in \mathbb{F}_2^m \\ M & \begin{pmatrix} & & \\ & M(\mathbf{v}) & \\ & & \end{pmatrix} \end{array} := E(m,r)$$

(Every codeword is spanned by the rows.)

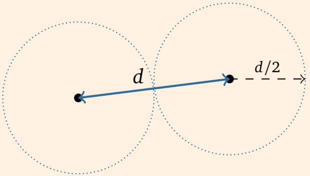
Linear codes can be also defined by their **parity check matrix**: a matrix H whose kernel is the code.

Duality: $RM(m,r)^{\perp} = RM(m, m-r-1)$.

$\implies$ parity check matrix of $RM(m,r)$ is $E(m, m-r-1)$.

DECODING REED-MULLER CODES

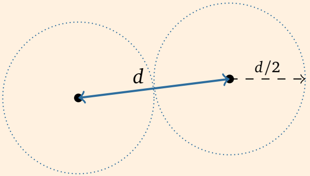
Worst Case Errors: Up to $d/2$ (d is minimal distance).



(algorithm by Reed54)

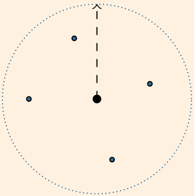
Decoding Reed-Muller Codes

Worst Case Errors: Up to $d/2$ (d is minimal distance).



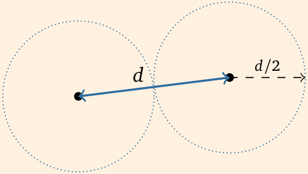
(algorithm by Reed54)

List Decoding: max radius with constant # of words



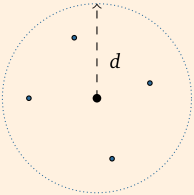
DECODING REED-MULLER CODES

Worst Case Errors: Up to $d/2$ (d is minimal distance).



(algorithm by **Reed54**)

List Decoding: max radius with constant # of words



Gopalan-Klivans-Zuckerman08, Bhowmick-Lovett15:

List decoding radius = d.

Decoding Reed-Muller Codes: Average Case

- Reed-Muller Codes are not very good with respect to **worst-case** errors (can't have constant rate and min distance at the same time)

Decoding Reed-Muller Codes: Average Case

- Reed-Muller Codes are not very good with respect to **worst-case** errors (can't have constant rate and min distance at the same time)
- What about the Shannon model of **random** corruptions?

Decoding Reed-Muller Codes: Average Case

- Reed-Muller Codes are not very good with respect to **worst-case** errors (can't have constant rate and min distance at the same time)
- What about the Shannon model of **random** corruptions?
- Standard model in coding theory with recent breakthroughs in the last few years (e.g. Arıkan's polar codes)

Decoding Reed-Muller Codes: Average Case

- Reed-Muller Codes are not very good with respect to **worst-case** errors (can't have constant rate and min distance at the same time)
- What about the Shannon model of **random** corruptions?
- Standard model in coding theory with recent breakthroughs in the last few years (e.g. Arıkan's polar codes)
- An ongoing research endeavor: how do Reed-Muller perform in Shannon's random error model?

MODELS FOR RANDOM CORRUPTIONS (CHANNELS)

- Binary Erasure Channel — $\mathrm{BEC}(p)$
  - Each bit independently replaced by '?' with probability p

MODELS FOR RANDOM CORRUPTIONS (CHANNELS)

Binary Erasure Channel — $\mathrm{BEC}(p)$

Each bit independently replaced by '?' with probability p

0	0	0	1	0	1	1	0

MODELS FOR RANDOM CORRUPTIONS (CHANNELS)

Binary Erasure Channel — $\mathrm{BEC}(p)$

Each bit independently replaced by '?' with probability p

0	?	0	1	?	1	?	0

MODELS FOR RANDOM CORRUPTIONS (CHANNELS)

Binary Erasure Channel — $\mathrm{BEC}(p)$

Each bit independently replaced by '?' with probability p

0	?	0	1	?	1	?	0

Binary Symmetric Channel — $\mathrm{BSC}(p)$

Each bit independently flipped with probability p

MODELS FOR RANDOM CORRUPTIONS (CHANNELS)

Binary Erasure Channel — $\mathrm{BEC}(p)$

Each bit independently replaced by '?' with probability p

0	?	0	1	?	1	?	0

Binary Symmetric Channel — $\mathrm{BSC}(p)$

Each bit independently flipped with probability p

0	0	0	1	0	1	1	0

MODELS FOR RANDOM CORRUPTIONS (CHANNELS)

Binary Erasure Channel — $\mathrm{BEC}(p)$

Each bit independently replaced by '?' with probability p

0	?	0	1	?	1	?	0

Binary Symmetric Channel — $\mathrm{BSC}(p)$

Each bit independently flipped with probability p

0	1	0	1	1	1	0	0

MODELS FOR RANDOM CORRUPTIONS (CHANNELS)

Binary Erasure Channel — $\mathrm{BEC}(p)$

Each bit independently replaced by '?' with probability p

0	?	0	1	?	1	?	0

Binary Symmetric Channel — $\mathrm{BSC}(p)$

Each bit independently flipped with probability p

0	1	0	1	1	1	0	0

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

MODELS FOR RANDOM CORRUPTIONS (CHANNELS)

Binary Erasure Channel — $\mathrm{BEC}(p)$

Each bit independently replaced by '?' with probability p

0	?	0	1	?	1	?	0

Binary Symmetric Channel — $\mathrm{BSC}(p)$

Each bit independently flipped with probability p

0	1	0	1	1	1	0	0

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

Shannon48: max rate that enables decoding (w.h.p.) is $1 - p$ (for BEC) and $1 - H(p)$ (for BSC). Codes achieving 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

Decoding Erasures

- How many random erasures can $RM(m, m-r-1)$ tolerate?

DECODING ERASURES

- How many random erasures can $RM(m, m-r-1)$ tolerate?
- **Fact:** for any linear code, a subset $S \subseteq [n]$ of erasures is uniquely decodable $\iff$ columns indexed by S in parity-check matrix are linearly independent
  decoding algo: solve system of linear equations

DECODING ERASURES

- How many random erasures can $RM(m, m-r-1)$ tolerate?
- **Fact:** for any linear code, a subset $S \subseteq [n]$ of erasures is uniquely decodable $\iff$ columns indexed by S in parity-check matrix are linearly independent
  decoding algo: solve system of linear equations
- **Reminder:** PCM of $RM(m, m-r-1)$ is $E(m, r)$

DECODING ERASURES

- How many random erasures can $RM(m, m-r-1)$ tolerate?
- **Fact:** for any linear code, a subset $S \subseteq [n]$ of erasures is uniquely decodable $\iff$ columns indexed by S in parity-check matrix are linearly independent
  decoding algo: solve system of linear equations
- **Reminder:** PCM of $RM(m, m-r-1)$ is $E(m, r)$
- $\implies$ can't correct more than $\binom{m}{\leq r}$ erasures

DECODING ERASURES

- How many random erasures can $RM(m, m-r-1)$ tolerate?
- **Fact:** for any linear code, a subset $S \subseteq [n]$ of erasures is uniquely decodable $\iff$ columns indexed by S in parity-check matrix are linearly independent
  decoding algo: solve system of linear equations
- **Reminder:** PCM of $RM(m, m-r-1)$ is $E(m, r)$
- $\implies$ can't correct more than $\binom{m}{\leq r}$ erasures
  (also follows from Shannon's theorem)
- **Question:** Can we correct $(1-o(1))\binom{m}{\leq r}$ erasures?

DECODING ERASURES

- How many random erasures can $RM(m, m-r-1)$ tolerate?
- **Fact:** for any linear code, a subset $S \subseteq [n]$ of erasures is uniquely decodable $\iff$ columns indexed by S in parity-check matrix are linearly independent
  decoding algo: solve system of linear equations
- **Reminder:** PCM of $RM(m, m-r-1)$ is $E(m, r)$
- $\implies$ can't correct more than $\binom{m}{\leq r}$ erasures
  (also follows from Shannon's theorem)
- **Question:** Can we correct $(1-o(1))\binom{m}{\leq r}$ erasures?
- if yes, $RM(m, m-r-1)$ **achieves capacity** for the BEC

REED-MULLER CODES AND THE BEC

For $\mathbf{v} \in \mathbb{F}_2^m$, let $\mathbf{v}^r =$ the column indexed by $\mathbf{v}$ in $E(m, r)$

(all evals of monoms of $\mathbf{deg} \leq r$ on $\mathbf{v}$)

Reed-Muller Codes and the BEC

For $\mathbf{v} \in \mathbb{F}_2^m$, let $\mathbf{v}^r =$ the column indexed by $\mathbf{v}$ in $E(m,r)$
(all evals of monoms of $\mathbf{deg} \leq r$ on $\mathbf{v}$)

What's the max t such that for
randomly picked $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathbb{F}_2^m$,
$\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r$ are linearly independent?

REED-MULLER CODES AND THE BEC

For $\mathbf{v} \in \mathbb{F}_2^m$, let $\mathbf{v}^r =$ the column indexed by $\mathbf{v}$ in $E(m,r)$
(all evals of monoms of $\mathbf{deg} \leq r$ on $\mathbf{v}$)

What's the max t such that for randomly picked $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathbb{F}_2^m$, $\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r$ are linearly independent?

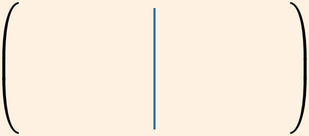
$$\underbrace{\begin{pmatrix} \quad & \quad & \quad \\ \quad & \quad & \quad \\ \quad & \quad & \quad \end{pmatrix}}_{E(m,r)}$$

REED-MULLER CODES AND THE BEC

For $\mathbf{v} \in \mathbb{F}_2^m$, let $\mathbf{v}^r =$ the column indexed by $\mathbf{v}$ in $E(m,r)$

(all evals of monoms of $\mathbf{deg} \leq r$ on $\mathbf{v}$)

What's the max t such that for randomly picked $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathbb{F}_2^m$, $\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r$ are linearly independent?

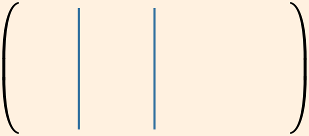


$E(m,r)$

REED-MULLER CODES AND THE BEC

For $\mathbf{v} \in \mathbb{F}_2^m$, let $\mathbf{v}^r =$ the column indexed by $\mathbf{v}$ in $E(m, r)$
(all evals of monoms of $\mathbf{deg} \leq r$ on $\mathbf{v}$)

What's the max t such that for randomly picked $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathbb{F}_2^m$, $\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r$ are linearly independent?

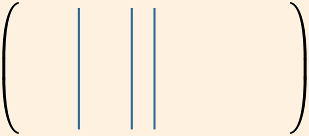


$E(m, r)$

REED-MULLER CODES AND THE BEC

For $\mathbf{v} \in \mathbb{F}_2^m$, let $\mathbf{v}^r$ = the column indexed by $\mathbf{v}$ in $E(m, r)$
(all evals of monoms of $\mathbf{deg} \leq r$ on $\mathbf{v}$)

What's the max t such that for randomly picked $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathbb{F}_2^m$, $\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r$ are linearly independent?



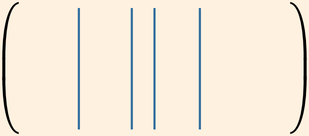
$E(m, r)$

REED-MULLER CODES AND THE BEC

For $\mathbf{v} \in \mathbb{F}_2^m$, let $\mathbf{v}^r =$ the column indexed by $\mathbf{v}$ in $E(m, r)$

(all evals of monoms of $\mathbf{deg} \leq r$ on $\mathbf{v}$)

What's the max t such that for randomly picked $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathbb{F}_2^m$, $\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r$ are linearly independent?

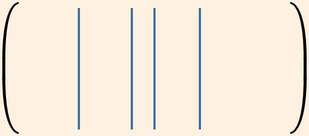


$E(m, r)$

REED-MULLER CODES AND THE BEC

For $\mathbf{v} \in \mathbb{F}_2^m$, let $\mathbf{v}^r =$ the column indexed by $\mathbf{v}$ in $E(m,r)$
(all evals of monoms of $\mathbf{deg} \le r$ on $\mathbf{v}$)

What's the max t such that for randomly picked $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathbb{F}_2^m$, $\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r$ are linearly independent?



$E(m,r)$

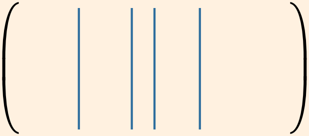
If $t = (1 - o(1))\binom{m}{\le r}$, $RM(m, m-r-1)$ achieves capacity for BEC.

REED-MULLER CODES AND THE BEC

For $\mathbf{v} \in \mathbb{F}_2^m$, let $\mathbf{v}^r =$ the column indexed by $\mathbf{v}$ in $E(m,r)$

(all evals of monoms of $\mathbf{deg} \leq r$ on $\mathbf{v}$)

What's the max t such that for randomly picked $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathbb{F}_2^m$, $\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r$ are linearly independent?



$E(m,r)$

If $t = (1-o(1))\binom{m}{\leq r}$, $RM(m, m-r-1)$ achieves capacity for BEC.

sanity check: $r = 1$ is good

Reed-Muller Codes and the BEC

Main Question: Does $RM(m, r)$ achieve capacity for BEC?

REED-MULLER CODES AND THE BEC

Main Question: Does $RM(m,r)$ achieve capacity for BEC?

Abbe-Shpilka-Wigderson15: YES if $r = o(m)$ or $r = m - o(\sqrt{m/\log m})$.

REED-MULLER CODES AND THE BEC

Main Question: Does $RM(m,r)$ achieve capacity for BEC?

Abbe-Shpilka-Wigderson15: YES if $r = o(m)$ or $r = m - o(\sqrt{m/\log m})$.

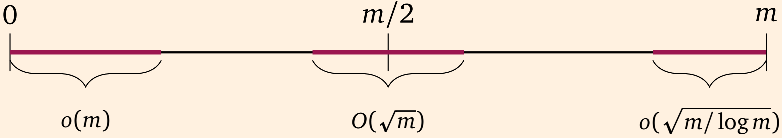
Kumar-Pfister15, Kudekar-Mondelli-Şaşoğlu-Urbanke15: YES if rate is constant (i.e. $r = m/2 \pm O(\sqrt{m})$).

REED-MULLER CODES AND THE BEC

Main Question: Does $RM(m, r)$ achieve capacity for BEC?

Abbe-Shpilka-Wigderson15: YES if $r = o(m)$ or $r = m - o(\sqrt{m/\log m})$.

Kumar-Pfister15, Kudekar-Mondelli-Şaşoğlu-Urbanke15: YES if rate is constant (i.e. $r = m/2 \pm O(\sqrt{m})$).

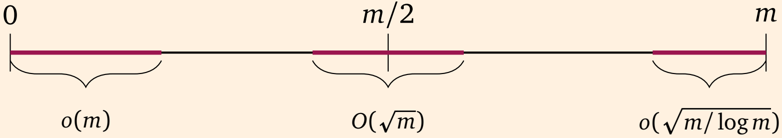


REED-MULLER CODES AND THE BEC

Main Question: Does $RM(m,r)$ achieve capacity for BEC?

Abbe-Shpilka-Wigderson15: YES if $r = o(m)$ or $r = m - o(\sqrt{m/\log m})$.

Kumar-Pfister15, Kudekar-Mondelli-Şaşoğlu-Urbanke15: YES if rate is constant (i.e. $r = m/2 \pm O(\sqrt{m})$).



Open Problem: Prove for every degree r.

Decoding erasures to decoding errors

How is this relevant for **error** correction?

DECODING ERASURES TO DECODING ERRORS

How is this relevant for **error** correction?

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

Decoding erasures to decoding errors

How is this relevant for **error** correction?

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

This talk: There is an efficient algorithm that corrects from errors, in $RM(m, m-2r-2)$, any pattern which is correctable from erasures in $RM(m, m-r-1)$.

Decoding Erasures to Decoding Errors

How is this relevant for **error** correction?

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

This talk: There is an efficient algorithm that corrects from errors, in $RM(m, m-2r-2)$, any pattern which is correctable from erasures in $RM(m, m-r-1)$.

(+ extensions to larger alphabets and other codes)

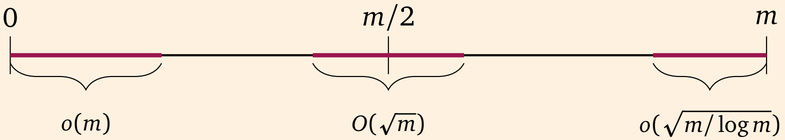
DECODING ERASURES TO DECODING ERRORS

How is this relevant for **error** correction?

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

This talk: There is an efficient algorithm that corrects from errors, in $RM(m, m-2r-2)$, any pattern which is correctable from erasures in $RM(m, m-r-1)$.

(+ extensions to larger alphabets and other codes)



DECODING ERRORS IN RM CODES

Corollary #1: (low-rate) efficient decoding algo for $(\frac{1}{2} - o(1))n$ random errors in $RM(m, o(\sqrt{m}))$ (min distance is $2^{m-\sqrt{m}}$) .

DECODING ERRORS IN RM CODES

Corollary #1: (low-rate) efficient decoding algo for $(\frac{1}{2} - o(1))n$ random errors in $RM(m, o(\sqrt{m}))$ (min distance is $2^{m-\sqrt{m}}$) .

Corollary #2: (high-rate) efficient decoding algo for $(1 - o(1))\binom{m}{\leq r}$ random errors in $RM(m, m-r)$ if $r = o(\sqrt{m/\log m})$ (min distance is 2^r) .

DECODING ERRORS IN RM CODES

Corollary #1: (low-rate) efficient decoding algo for $(\frac{1}{2} - o(1))n$ random errors in $RM(m, o(\sqrt{m}))$ (min distance is $2^{m-\sqrt{m}}$) .

Corollary #2: (high-rate) efficient decoding algo for $(1 - o(1))\binom{m}{\leq r}$ random errors in $RM(m, m-r)$ if $r = o(\sqrt{m/\log m})$ (min distance is 2^r) .

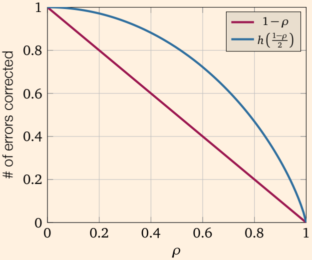
Corollary #3: If $RM(m, \left(\frac{1+\rho}{2}\right)m)$ achieves capacity, efficient decoding algo for $2^{h\left(\frac{1-\rho}{2}\right)m}$ random errors in $RM(m, \rho m)$.

DECODING ERRORS IN RM CODES

Corollary #1: (low-rate) efficient decoding algo for $(\frac{1}{2} - o(1))n$ random errors in $RM(m, o(\sqrt{m}))$ (min distance is $2^{m-\sqrt{m}}$) .

Corollary #2: (high-rate) efficient decoding algo for $(1-o(1))\binom{m}{\leq r}$ random errors in $RM(m, m-r)$ if $r = o(\sqrt{m/\log m})$ (min distance is 2^r) .

Corollary #3: If $RM(m, \left(\frac{1+\rho}{2}\right)m)$ achieves capacity, efficient decoding algo for $2^{h\left(\frac{1-\rho}{2}\right)m}$ random errors in $RM(m, \rho m)$.



PROOF IDEA

Goal: decode in $RM(m, m-2r-2)$ every pattern which is correctable from erasures in $RM(m, m-r-1)$.

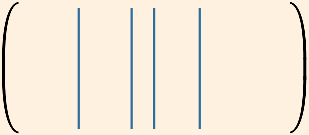
0	1	0	1	1	1	0	0

PROOF IDEA

Goal: decode in $RM(m, m-2r-2)$ every pattern which is correctable from erasures in $RM(m, m-r-1)$.

0	1	0	1	1	1	0	0

recall: $\{\mathbf{u}_1, \ldots, \mathbf{u}_t\}$ correctable from erasures iff $\{\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r\}$ are linearly independent.



$E(m,r)$

DUAL POLYNOMIALS

Fact: If $\{\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r\}$ lin. indep., $\exists$ polys $\{f_1, \ldots, f_t\}$ of $\deg \leq r$ such that

$$f_i(\mathbf{u}_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases}$$

DUAL POLYNOMIALS

Fact: If $\{\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r\}$ lin. indep., $\exists$ polys $\{f_1, \ldots, f_t\}$ of $\deg \leq r$ such that

$$f_i(\mathbf{u}_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases}$$

Proof:

$$\begin{pmatrix} \text{—} & \mathbf{u}_1^r & \text{—} \\ & \vdots & \\ \text{—} & \mathbf{u}_t^r & \text{—} \end{pmatrix} \cdot \begin{pmatrix} | \\ f_i \\ | \end{pmatrix} = \mathbf{e}_i$$

DUAL POLYNOMIALS

Fact: If $\{\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r\}$ lin. indep., $\exists$ polys $\{f_1, \ldots, f_t\}$ of $\deg \leq r$ such that

$$f_i(\mathbf{u}_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases}$$

Proof:

$$\begin{pmatrix} \text{---} & \mathbf{u}_1^r & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{u}_t^r & \text{---} \end{pmatrix} \cdot \begin{pmatrix} | \\ f_i \\ | \end{pmatrix} = \mathbf{e}_i$$

Solve this system for f_i.

DUAL POLYNOMIALS

Fact: If $\{\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r\}$ lin. indep., $\exists$ polys $\{f_1, \ldots, f_t\}$ of $\deg \leq r$ such that

$$f_i(\mathbf{u}_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases}$$

Proof:

$$\begin{pmatrix} \text{---} & \mathbf{u}_1^r & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{u}_t^r & \text{---} \end{pmatrix} \cdot \begin{pmatrix} | \\ f_i \\ | \end{pmatrix} = \mathbf{e}_i$$

Solve this system for f_i.

Our approach would be to **find** those polynomials.

ALGORITHM: SOLVE A SYSTEM OF LINEAR EQUATIONS

$U = \{\mathbf{u}_1, \ldots, \mathbf{u}_t\} \in \mathbb{F}_2^m$ (unknown) set of error locations.

ALGORITHM: SOLVE A SYSTEM OF LINEAR EQUATIONS

$U = \{\mathbf{u}_1, \ldots, \mathbf{u}_t\} \in \mathbb{F}_2^m$ (unknown) set of error locations.

Main Claim: For every $\mathbf{v} \in \{0,1\}^m$, $\mathbf{v} \in U \iff$ the following linear system, in unknown coeffs of degree r poly f, is solvable:

Algorithm: Solve a System of Linear Equations

$U = \{\mathbf{u}_1, \ldots, \mathbf{u}_t\} \in \mathbb{F}_2^m$ (unknown) set of error locations.

Main Claim: For every $\mathbf{v} \in \{0,1\}^m$, $\mathbf{v} \in U \iff$ the following linear system, in unknown coeffs of degree r poly f, is solvable:

$\forall$ monomial M, $\deg M \leq r$:

ALGORITHM: SOLVE A SYSTEM OF LINEAR EQUATIONS

$U = \{\mathbf{u}_1, \ldots, \mathbf{u}_t\} \in \mathbb{F}_2^m$ (unknown) set of error locations.

Main Claim: For every $\mathbf{v} \in \{0,1\}^m$, $\mathbf{v} \in U \iff$ the following linear system, in unknown coeffs of degree r poly f, is solvable:

$\forall$ monomial M, $\deg M \leq r$:

$$\sum_{i=1}^{t} f(\mathbf{u}_i) = f(\mathbf{v}) = 1$$

(f non-trivial)

Algorithm: Solve a System of Linear Equations

$U = \{\mathbf{u}_1, \ldots, \mathbf{u}_t\} \in \mathbb{F}_2^m$ (unknown) set of error locations.

Main Claim: For every $\mathbf{v} \in \{0,1\}^m$, $\mathbf{v} \in U \iff$ the following linear system, in unknown coeffs of degree r poly f, is solvable:

$\forall$ monomial M, $\deg M \leq r$:

$$\sum_{i=1}^{t} f(\mathbf{u}_i) = f(\mathbf{v}) = 1 \qquad (f \text{ non-trivial})$$

$$\sum_{i=1}^{t} (f \cdot M)(\mathbf{u}_i) = M(\mathbf{v}) \qquad (\mathbf{v} \text{ spanned by } U)$$

ALGORITHM: SOLVE A SYSTEM OF LINEAR EQUATIONS

$U = \{\mathbf{u}_1, \ldots, \mathbf{u}_t\} \in \mathbb{F}_2^m$ (unknown) set of error locations.

Main Claim: For every $\mathbf{v} \in \{0,1\}^m$, $\mathbf{v} \in U \iff$ the following linear system, in unknown coeffs of degree r poly f, is solvable:

$\forall$ monomial M, $\deg M \leq r$:

$$\sum_{i=1}^{t} f(\mathbf{u}_i) = f(\mathbf{v}) = 1 \qquad (f \text{ non-trivial})$$

$$\sum_{i=1}^{t} (f \cdot M)(\mathbf{u}_i) = M(\mathbf{v}) \qquad (\mathbf{v} \text{ spanned by } U)$$

$$\sum_{i=1}^{t} (f \cdot M \cdot (x_\ell + \mathbf{v}_\ell + 1))(\mathbf{u}_i) = M(\mathbf{v})$$

($\mathbf{v}$ spanned by vectors in U that agree with its ℓ-th coordinate)

ALGORITHM: SOLVE A SYSTEM OF LINEAR EQUATIONS

$U = \{\mathbf{u}_1, \ldots, \mathbf{u}_t\} \in \mathbb{F}_2^m$ (unknown) set of error locations.

Main Claim: For every $\mathbf{v} \in \{0,1\}^m$, $\mathbf{v} \in U \iff$ the following linear system, in unknown coeffs of degree r poly f, is solvable:

$\forall$ monomial M, $\deg M \leq r$:

$$\sum_{i=1}^{t} f(\mathbf{u}_i) = f(\mathbf{v}) = 1$$ (f non-trivial)

$$\sum_{i=1}^{t} (f \cdot M)(\mathbf{u}_i) = M(\mathbf{v})$$ ($\mathbf{v}$ spanned by U)

$$\sum_{i=1}^{t} (f \cdot M \cdot (x_\ell + \mathbf{v}_\ell + 1))(\mathbf{u}_i) = M(\mathbf{v})$$

($\mathbf{v}$ spanned by vectors in U that agree with its ℓ-th coordinate)

If U lin. indep. and $\mathbf{v} = \mathbf{u}_i \in U$, f_i is a solution. Conversely, if solvable and U lin. indep., can show $\mathbf{v} \in U$.

SETTING UP SYSTEM OF EQUATIONS

Every coefficient in the system is of the form $\sum_{i=1}^{t} g(\mathbf{u}_i)$ for poly g of degree $\leq 2r+1$.

SETTING UP SYSTEM OF EQUATIONS

Every coefficient in the system is of the form $\sum_{i=1}^{t} g(\mathbf{u}_i)$ for poly g of degree $\leq 2r+1$.

How to compute the coefficients?

SETTING UP SYSTEM OF EQUATIONS

Every coefficient in the system is of the form $\sum_{i=1}^{t} g(\mathbf{u}_i)$ for poly g of degree $\leq 2r+1$.

How to compute the coefficients?

Input to the algo: $\mathbf{y} = \mathbf{c} + \mathbf{e}$ with $\mathbf{c} \in RM(m, m-2r-2)$, and $\mathbf{e}$ characteristic vector of $U = \{\mathbf{u}_1, \ldots, \mathbf{u}_t\}$.

SETTING UP SYSTEM OF EQUATIONS

Every coefficient in the system is of the form $\sum_{i=1}^{t} g(\mathbf{u}_i)$ for poly g of degree $\leq 2r+1$.

How to compute the coefficients?

Input to the algo: $\mathbf{y} = \mathbf{c} + \mathbf{e}$ with $\mathbf{c} \in RM(m, m-2r-2)$, and $\mathbf{e}$ characteristic vector of $U = \{\mathbf{u}_1, \ldots, \mathbf{u}_t\}$.

Syndrome of $\mathbf{y}$: $E(m, 2r+1) \cdot \mathbf{y} = E(m, 2r+1) \cdot \mathbf{e}$.

(recall: $E(m, 2r+1)$ is PCM of $RM(m, m-2r-2)$)

SETTING UP SYSTEM OF EQUATIONS

Every coefficient in the system is of the form $\sum_{i=1}^{t} g(\mathbf{u}_i)$ for poly g of degree $\leq 2r+1$.

How to compute the coefficients?

Input to the algo: $\mathbf{y} = \mathbf{c} + \mathbf{e}$ with $\mathbf{c} \in RM(m, m-2r-2)$, and $\mathbf{e}$ characteristic vector of $U = \{\mathbf{u}_1, \ldots, \mathbf{u}_t\}$.

Syndrome of $\mathbf{y}$: $E(m, 2r+1) \cdot \mathbf{y} = E(m, 2r+1) \cdot \mathbf{e}$.

(recall: $E(m, 2r+1)$ is PCM of $RM(m, m-2r-2)$)

Corollary: The syndrome of $\mathbf{y}$ is a $\binom{m}{\leq 2r+1}$ long vector α, where $\alpha_M = \sum_{i=1}^{t} M(\mathbf{u}_i)$, for every monom M, $\deg M \leq 2r+1$.

BACK TO BEC

The algorithm works whenever $\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r$ lin. indep., which happens (w.h.p.) whenever $RM(m, m-r-1)$ achieves capacity.

BACK TO BEC

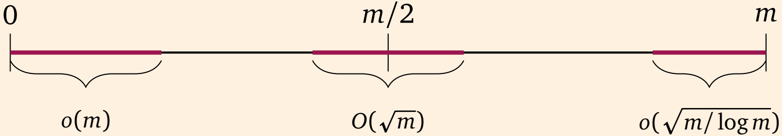
The algorithm works whenever $\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r$ lin. indep., which happens (w.h.p.) whenever $RM(m, m-r-1)$ achieves capacity.

Restating the main question: for which values of r, $\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r$ are linearly independent with high probability for $t = (1-o(1))\binom{m}{\leq r}$?

BACK TO BEC

The algorithm works whenever $\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r$ lin. indep., which happens (w.h.p.) whenever $RM(m, m-r-1)$ achieves capacity.

Restating the main question: for which values of r, $\mathbf{u}_1^r, \ldots, \mathbf{u}_t^r$ are linearly independent with high probability for $t = (1 - o(1))\binom{m}{\leq r}$?



SUMMARY

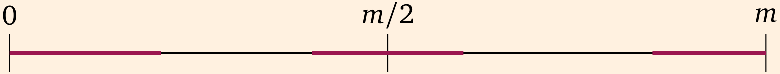
Decoding algo for RM far beyond the minimal distance, both for high-rate and low-rate regimes.

SUMMARY

Decoding algo for RM far beyond the minimal distance, both for high-rate and low-rate regimes.

Open Problems:

- Prove RM achieves capacity for BEC for all degrees

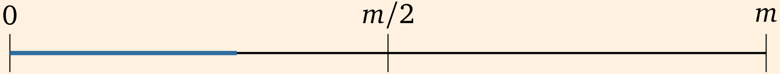


SUMMARY

Decoding algo for RM far beyond the minimal distance, both for high-rate and low-rate regimes.

Open Problems:

- Prove RM achieves capacity for BEC for all degrees
- RM for BSC: much less is known! (**ASW** proved it achieves capacity for small rates)

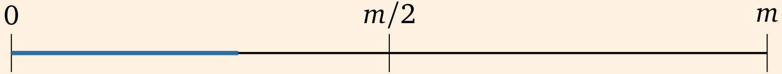


SUMMARY

Decoding algo for RM far beyond the minimal distance, both for high-rate and low-rate regimes.

Open Problems:

- Prove RM achieves capacity for BEC for all degrees
- RM for BSC: much less is known! (**ASW** proved it achieves capacity for small rates)



THANK YOU