Reed-Muller codes for random errors and erasures

Based on:

Abbe-S-Wigderson
Saptharishi-S-Volk
Reed-Muller code - $\text{RM}(m,r)$

- **Message:** coefficient vector of a polynomial $f(x_1,\ldots,x_m)$ over $\mathbb{F}_2$ of degree $\leq r$

- **Encoding:** evaluations of $f$:
  
  $f \rightarrow (f(000), f(001), \ldots, f(111))$

- **Distance:** Hamming distance between any two code words is $\geq 2^{m-r}$

- **Note:** codewords form a linear space over $\mathbb{F}_2$
The Sierpinski matrix

\[
\begin{pmatrix}
1, x_1, x_2, x_1x_2, x_3, x_1x_3, \\
\vdots
\end{pmatrix}
\]

m-variate monomials in lexicographical order:

\[
\{0,1\}^m
\]
The Sierpinski matrix

\[ \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \]

\{0, 1\}^m

m-variate monomials in lexicographical order:

1, x_1, x_2, x_1x_2, x_3, x_1x_3, ...

Keep rows corresponding to low degree monomials (heaviest rows)
The Sierpinski matrix

\[
\begin{bmatrix}
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 
\end{bmatrix}
\]

Keep rows corresponding to low degree monomials (heaviest rows)
The Sierpinski matrix

Keep rows corresponding to low degree monomials (heaviest rows)
The Sierpinski Matrix

Generating matrix of Reed-Muller code:
codewords are linear combination of rows

Keep rows corresponding to low degree monomials
(heaviest rows)

Generating matrix of Reed-Muller code:
codewords are linear combination of rows
Reed-Muller code - \( \text{RM}(m,r) \)

- **Message:** coefficient vectors of a polynomial \( f(x_1,\ldots,x_m) \) over \( \mathbb{F}_2 \) of degree \( \leq r \)
- **Encoding:** evaluations of \( f \):
  \( f \rightarrow (f(000),f(001),\ldots,f(111)) \)
- **Minimum distance:** \( d=2^{m-r} \)
- Most studied linear algebraic code
- Around over 50 years yet fundamental questions are still open!
Why CS cares about RM codes?
Why CS cares about RM codes?

- Low degree polynomials ubiquitous in TCS:
  - lower bounds
  - derandomization
  - PCP
  - Hardness amplification
  - List decoding
  - Algorithms
  - Property testing
  - Extractors
  - ...
Decoding RM codes

Decoding problem: decode corrupted codewords

Worst case behavior well understood:

Reed: Efficient decoding up to half min. distance

Gopalan-Klivans-Zuckerman, Bhowmick-Lovett: List decoding radius $\leq 2 \cdot \text{dist}$
Decoding RM codes

- **Decoding problem**: decode corrupted codewords
- **Worst case** behavior well understood:
  - **Reed**: Efficient decoding up to half min. distance
  - **Gopalan-Klivans-Zuckerman, Bhowmick-Lovett**:
    List decoding radius \( \leq 2 \cdot \text{dist} \)
- Can we do better for random errors?
- **Average case** study of errors
Decoding RM codes

Decoding problem: decode corrupted codewords

Worst case behavior well understood:

Reed: Efficient decoding up to half min. distance

Gopalan-Klivans-Zuckerman, Bhowmick-Lovett:
List decoding radius $\leq 2 \cdot \text{dist}$

Can we do better for random errors?

Average case study of errors

Old open problem in coding theory:

How good are RM codes?

Do RM codes meet Shannon's bounds for random errors and erasures?
Error Model ⟷ Channels

- **Binary Erasure Channel:**
  symbol is replaced with ? (erased) with probability $p$
Error Model ⟷ Channels

- **Binary Erasure Channel:**
  symbol is replaced with ? (erased) with probability p

```
0 1 1 0 1 0 0 0 1 0
```
Error Model ↔ Channels

Binary Erasure Channel:
symbol is replaced with ? (erased) with probability p
Error Model $\leftrightarrow$ Channels

- **Binary Erasure Channel:**
  symbol is replaced with ? (erased) with probability $p$

- **Binary Symmetric Channel:**
  symbol is flipped with probability $p$
Error Model ↔ Channels

- **Binary Erasure Channel:**
  symbol is replaced with ? (erased) with probability $p$

- **Binary Symmetric Channel:**
  symbol is flipped with probability $p$
Error Model ↔ Channels

- **Binary Erasure Channel:**
  symbol is replaced with ? (erased) with probability $p$

- **Binary Symmetric Channel:**
  symbol is flipped with probability $p$

![Binary Symmetric Channel Diagram]

```
0 1 1 0 1 0 0 0 1 0
```

```
0 1 0 1 1 1 0 1 0 1 0
```
Error Model ↔ Channels

- **Binary Erasure Channel:**
  symbol is replaced with ? (erased) with probability $p$

- **Binary Symmetric Channel:**
  symbol is flipped with probability $p$
Error Model ↔ Channels

- **Binary Erasure Channel:** symbol is replaced with ? (erased) with probability $p$
- **Binary Symmetric Channel:** symbol is flipped with probability $p$
- **Shannon:** maximal rate that enables decoding w.h.p. (capacity of channel) - best tradeoff between redundancy and robustness
  - BEC: $R = 1 - p$
  - BSC $R = 1 - h(p)$ ($h(x) = -x \log(x) - (1-x) \log(1-x)$)
Error Model ↔ Channels

- **Binary Erasure Channel:** symbol is replaced with ? (erased) with probability $p$
- **Binary Symmetric Channel:** symbol is flipped with probability $p$

**Shannon:** maximal rate that enables decoding w.h.p. (capacity of channel) - best tradeoff between redundancy and robustness

- **BEC:** $R = 1 - p$
- **BSC** $R = 1 - h(p)$ (where $h(x) = -x\log(x) - (1-x)\log(1-x)$)

**Major goal:** design explicit codes that meet Shannon’s bound (with efficient encoding and decoding)
Average case behavior of RM

Do Reed-Muller codes meet Shannon’s bound? I.e., can RM codes of rate R handle the same fraction of errors/erasures that random codes of rate R handle.
Average case behavior of RM

Do Reed-Muller codes meet Shannon’s bound?
I.e., can RM codes of rate R handle the same fraction of errors/erasures that random codes of rate R handle.

Problem related to

- Rank of random evaluation matrices
- Spaces of tensors over $\mathbb{F}_2$
- Polynomial interpolation from noisy data
- Sparse recovery
- Learning
- Polar codes
- …
Polar Codes

Introduced by Arikan 2009

Very similar to RM - messages are polynomials with respect to a different monomial basis (different choice of rows from the Serpiensky matrix)

Achieve capacity for all channels!

Due to strange basis (no simple description), Polar codes seem less natural than RM codes, yet are still morally similar

It would be much nicer to work with RM codes rather than Polar!
## Results

<table>
<thead>
<tr>
<th>BEC</th>
<th>BSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r = o(m)$</td>
<td>✓</td>
</tr>
<tr>
<td>$r = m - O\left(\sqrt{m / \log(m)}\right)$</td>
<td>✓</td>
</tr>
</tbody>
</table>

- Note: minimum distance of RM(m,m-s) is $2^s$
Results

<table>
<thead>
<tr>
<th></th>
<th>BEC</th>
<th>BSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r = o(m)$</td>
<td>$\sqrt{}$</td>
<td>$\sqrt{}$</td>
</tr>
<tr>
<td>$r = m - O(\sqrt{m / \log(m)})$</td>
<td>$\sqrt{}$</td>
<td>$\sqrt{}$</td>
</tr>
<tr>
<td></td>
<td>$\sqrt{\text{(# of errors)}} \approx \sqrt{\left\lceil \frac{m}{r} \right\rceil}$</td>
<td></td>
</tr>
</tbody>
</table>

- Note: minimum distance of $\text{RM}(m, m-s)$ is $2^s$
- Prior work: $r=1$ (folklore). $r=2$ (Helleseth, Klove, Levenshtein ’05)
# Results

<table>
<thead>
<tr>
<th>BEC</th>
<th>BSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>(r = o(m)) &amp; (\sqrt{\text{# of errors}})</td>
<td></td>
</tr>
<tr>
<td>(r = m - O\left(\sqrt{m / \log(m)}\right)) &amp; (\sqrt{\left\lceil \frac{m}{r} \right\rceil} \approx m / \log(m))</td>
<td></td>
</tr>
</tbody>
</table>

- Note: minimum distance of \(\text{RM}(m,m-s)\) is \(2^s\)
- Prior work: \(r=1\) (folklore). \(r=2\) (Helleseth, Klove, Levenshtein ’05)
- Kumar-Pfister, Kudekar-Mondelli-Sasoglu-Urbanke 15: RM achieve capacity for the BEC at constant Rate!
Figure 1: Regime of $r$ for which $RM(m, r)$ is known to achieve capacity for the BEC
Our results:  
\[
\approx n/2 \text{ errors } \quad O(n^4) \text{ time algo. } \quad 2^h\left(\frac{1-e}{r}\right)^m \text{ errors for } \rho = r/m \quad (d \log d)^{O(\log m)} \text{ errors assuming RM achieve capacity for all rates } \quad n^{1+o(1)} \text{ time algo.}
\]

[Dum04, DS06, Dum06]:  
\[
\approx n/2 \text{ errors } \quad O(d \log d) = O(2^{(1-\rho)m} \cdot (1-\rho)m) \text{ errors for } \rho = r/m \quad O(n \log n) \text{ time algorithm}
\]

Degree (\(r\)) of RM(\(m, r\)):  
\[
0 \quad o(\sqrt{m}) \quad m/2 \quad m
\]
\[
\log m \quad o(\sqrt{m}) \quad m/2 \quad m
\]

Figure 1: Regime of \(r\) for which RM(\(m, r\)) is known to achieve capacity for the BEC

Figure 3: Comparison with [Dum04, DS06, Dum06]
BEC theorems
Reduction to rank of submatrices

Questions about BEC (erasures) boil down to

1. **High rate**: Is a random set of columns (slightly less than number of rows), of the generating matrix, linearly independent?

2. **Low rate**: Does a random set of columns (slightly more than number of rows), of the generating matrix, span all other columns?

We give positive answers which solve the BEC case for high and low rate, respectively.

Very different from Kumar-Pfister, Kudekar-Mondelli-Sasoglu-Urbanke
Generating Matrix of RM(4,2)
Generating Matrix of RM(4,2)

<table>
<thead>
<tr>
<th></th>
<th>0000</th>
<th>0001</th>
<th>0010</th>
<th>0011</th>
<th>0100</th>
<th>0101</th>
<th>0110</th>
<th>0111</th>
<th>1000</th>
<th>1001</th>
<th>1010</th>
<th>1011</th>
<th>1100</th>
<th>1101</th>
<th>1110</th>
<th>1111</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>x_1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>x_2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>x_3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>x_4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>x_1 x_2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>x_1 x_3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>x_1 x_4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>x_2 x_3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>x_2 x_4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>x_3 x_4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Is the set linearly independent?
Generating Matrix of RM(4,2)

Does the set span columns space?
Parity Check Matrix

- $H$ such that $v$ in Code iff $Hv = 0$
- $H$ is generating matrix for the dual code $C^\perp = \{u : u \perp v \text{ for all } v \text{ in } C\}$
- Fact $\text{RM}(m,r)^\perp = \text{RM}(m,m-r-1)$
- I.e., P.C.M is also an evaluation matrix (monomials vs. points)
Reduction (high-rate)

Want to solve linear system

$$\begin{bmatrix} 0 \\ ? \\ 1 \\ 1 \\ ? \\ 0 \\ ? \end{bmatrix} \times \begin{bmatrix} ? \\ ? \\ 1 \\ 1 \\ ? \\ 0 \\ ? \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$
Reduction (high-rate)

Want to solve linear system

\[
\begin{pmatrix}
0 \\
? \\
? \\
1 \\
1 \\
?
\end{pmatrix}
\times
PCM
\times
\begin{pmatrix}
0 \\
? \\
? \\
0 \\
0
\end{pmatrix}
= 
\begin{pmatrix}
0 \\
0 \\
0 \\
0 \\
0
\end{pmatrix}
\]

Unique solution (unique decoding) iff corresponding sub-matrix has full column-rank.
Reduction (low-rate)
Reduction (low-rate)
Reduction (low-rate)

msg \times \text{Gen}
Reduction (low-rate)
Reduction (low-rate)

\[ \text{msg} \times \text{Gen} = \begin{array}{ccccccccccc} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ \end{array} \]

\[ \begin{array}{ccccccccccc} 0 & 1 & ? & 0 & 1 & ? & 0 & 0 & 1 & 0 \end{array} \]
Reduction (low-rate)

Need to decode message based on non-erased information. Possible iff corresponding sub-matrix has full row-rank.
BSC theorems
Using the weight distribution
Using the weight distribution

Theorem 5: Tighter bounds on weight distribution of RM codes (based upon and improves Kaufman-Lovett-Porat)

#codewords of wt < $2^m/2^b \approx |RM(m,r)|^{(r/m)^b}$
Using the weight distribution

- **Theorem 5**: Tighter bounds on weight distribution of RM codes (based upon and improves Kaufman-Lovett-Porat)

\[ \#\text{codewords of wt} < 2^{m/2^b} \approx |RM(m,r)|^{(r/m)^b} \]

- Morally, weight distribution similar to random codes
Using the weight distribution

- **Theorem 5**: Tighter bounds on weight distribution of RM codes (based upon and improves Kaufman-Lovett-Porat)
  
  \[ \text{#codewords of wt } < \frac{2^m}{2^b} \approx |RM(m,r)|^{(r/m)^b} \]

- Morally, weight distribution similar to random codes

- **Theorem 2**: RM achieve capacity for BSC at low rate

  **Proof**: Bounds on weight distribution imply code achieves capacity (mimic proof that random codes achieve capacity)
Using the weight distribution

- **Theorem 5**: Tighter bounds on weight distribution of RM codes (based upon and improves Kaufman-Lovett-Porat)

  \[
  \text{#codewords of wt } < 2^{m/2^b} \approx |\text{RM}(m,r)|^{(r/m)^b}
  \]

- Morally, weight distribution similar to random codes

- **Theorem 2**: RM achieve capacity for BSC at low rate

- **Proof**: Bounds on weight distribution imply code achieves capacity (mimic proof that random codes achieve capacity)

- This connection was already observed by Poltyrev ‘94
Weight distribution

Theorem 5: Tighter bounds on weight distribution of RM codes (based upon and improves Kaufman-Lovett-Porat)

\[ \#\text{codewords of wt} < 2^{m/2^b} \approx |RM(m,r)|^{(r/m)^b} \]
Weight distribution

**Theorem 5:** Tighter bounds on weight distribution of RM codes (based upon and improves **Kaufman-Lovett-Porat**)

\[
\text{#codewords of wt } < 2^{m/2} \approx |RM(m,r)|^{(r/m)^b}
\]

**Idea of Proof:** f of small weight $\leftrightarrow$ f is a biased polynomial
Weight distribution

- **Theorem 5**: Tighter bounds on weight distribution of RM codes (based upon and improves Kaufman-Lovett-Porat)

  \#codewords of wt < \(2^m/2^b\) \(\approx |RM(m,r)|^{(r/m)b}\)

- Idea of Proof: f of small weight \(\leftrightarrow\) f is a biased polynomial

- \(\Rightarrow\) f very well approximated by few lower degree polynomials
Weight distribution

- **Theorem 5**: Tighter bounds on weight distribution of RM codes (based upon and improves Kaufman-Lovett-Porat)

\[
\text{#codewords of wt } < 2^m/2^b \approx |RM(m,r)|^{(r/m)^b}
\]

- **Idea of Proof**: f of small weight $\leftrightarrow$ f is a biased polynomial

  $\Rightarrow$ f very well approximated by few lower degree polynomials

  $\Rightarrow$ f computed by few lower degree polynomials
**Weight distribution**

- **Theorem 5**: Tighter bounds on weight distribution of RM codes (based upon and improves **Kaufman-Lovett-Porat**)

\[
\text{#codewords of } \text{wt} < 2^{m/2^b} \approx |\text{RM}(m,r)|^{(r/m)b}
\]

- **Idea of Proof**: \( f \) of small weight \( \leftrightarrow \) \( f \) is a biased polynomial

\[\Rightarrow f \text{ very well approximated by few lower degree polynomials}\]

\[\Rightarrow f \text{ computed by few lower degree polynomials}\]

- **KLP**: count number of such representations
Weight distribution

Theorem 5: Tighter bounds on weight distribution of RM codes (based upon and improves Kaufman-Lovett-Porat)

\[ \#\text{codewords of wt} < \frac{2^m}{2^b} \approx |\text{RM}(m,r)|^{\frac{r}{m}b} \]

Idea of Proof: \( f \) of small weight \( \leftrightarrow \) \( f \) is a biased polynomial

\( \Rightarrow f \) very well approximated by few lower degree polynomials

\( \Rightarrow f \) computed by few lower degree polynomials

KLP: count number of such representations

Improvement:

\( f \) somewhat approximated by fewer lower degree polynomials

Use recursion to bound the number of possible approximations
From erasures to errors

Theorem 4: RM decodable from “many” errors at high rate

Novel reduction from errors to erasures:
If the erasure pattern $U$ can be corrected in $\text{RM}(m, m-t)$ then the error pattern $U$ can be corrected in $\text{RM}(m, m-2t)$. 
From erasures to errors

Theorem 4: RM decodable from “many” errors at high rate

Novel reduction from errors to erasures:
If the erasure pattern $U$ can be corrected in $RM(m,m-t)$ then the error pattern $U$ can be corrected in $RM(m,m-2t)$.

Generalizes to any linear code
From erasures to errors

- **Theorem 4**: RM decodable from “many” errors at high rate

- **Novel reduction from errors to erasures**: If the *erasure pattern* $U$ can be corrected in $RM(m,m-t)$ then the *error pattern* $U$ can be corrected in $RM(m,m-2t)$.

- Generalizes to any linear code

- **Abbe-S-Wigderson**: result is “existential”, no efficient algorithm provided.
From erasures to errors

Theorem 4: RM decodable from “many” errors at high rate

Novel reduction from errors to erasures:
If the erasure pattern $U$ can be corrected in $\text{RM}(m,m-t)$ then the error pattern $U$ can be corrected in $\text{RM}(m,m-2t)$.

Generalizes to any linear code

Abbe-S-Wigderson: result is “existential”, no efficient algorithm provided.

Saptharishi-S-Volk: Efficient decoding algorithm
From erasures to errors

**Theorem 4:** RM decodable from “many” errors at high rate

**Novel reduction from errors to erasures:**
If the erasure pattern \( U \) can be corrected in \( \text{RM}(m, m-t) \) then the error pattern \( U \) can be corrected in \( \text{RM}(m, m-2t) \).

Generalizes to any linear code

**Abbe-S-Wigderson:** result is “existential”, no efficient algorithm provided.

**Saptharishi-S-Volk:** Efficient decoding algorithm

Result “boosted” by recent results of Kumar-Pfister, Kudekar-Mondelli-Sasoglu-Urbanke
Figure 1: Regime of $r$ for which $RM(m, r)$ is known to achieve capacity for the BEC
Our results:

- $\approx n/2$ errors
- $O(n^4)$ time algo.
- $2^h\left(\frac{1-\rho}{\rho}\right)^m$ errors for $\rho = r/m$
- Assuming $\text{RM}$ achieve capacity for all rates
- $(d \log d)^{O(\log m)}$ errors
- $n^{1+o(1)}$ time algo.

[Dum04, DS06, Dum06]:

- $\approx n/2$ errors
- $O(d \log d) = O(2^{(1-\rho)m} \cdot (1 - \rho)m)$ errors for $\rho = r/m$
- $O(n \log n)$ time algorithm

Degree ($r$) of $\text{RM}(m, r)$:

- $0$
- $o(\sqrt{m})$
- $m/2$
- $m$
- $\log m$
- $\log m$
- $o(\sqrt{m/\log m})$

(see also Figure 2)

Figure 3: Comparison with [Dum04, DS06, Dum06]
Summary

**Abbe-S-Wigderson**: RM codes achieve capacity for BEC for rates close to 0 or 1 and for BSC at rate close to 0

First improvement on a 50 years old problem

Several developments followed:

- **Kumar-Pfister, Kudekar-Mondelli-Sasoglu-Urbanke**: RM achieve BEC capacity for constant rate

- **Saptharishi-S-Volk**: efficient algorithm for correcting most error patterns of weight t in RM(m,m-2t)

**Open problem**: Do RM codes achieve capacity for the BSC?

**Most interesting case**: what can be said for the BSC for constant rate (degree m/2 ± O(√m))?

**Open problem**: Do RM codes achieve capacity for the BEC for all rates?