Bounded Single Insertion/Deletion Correcting Codes

Takayuki Nozaki

Yamaguchi University

July 11th, 2019
Overview

$r$-bounded single insertion/deletion correcting ($r$-BSIDC) code

- (Properties)
  - Correcting single insertion/deletion
  - (Assumption) receiver knows the range of positions occurring
    insertion/deletion (The range is $r$)
- (Application) Component codes for burst insertion/deletion correcting
  codes

Purpose of this research

Construction of $r$-BSIDC codes with large cardinality

Outline of this talk

1. Definitions and Examples
2. Existing codes (ST code, Shifted VT code)
3. Constructed codes (Exponential coefficient code, Odd coefficient code)
   (Construction, Cardinality, Decoding algorithm)
Definition and Example (1)

Definition: $r$-bounded single deletion correcting code

There exists a decoder which corrects a single deletion from the received sequence $\mathbf{y}$ and the range of deletion positions $[s, s + r - 1] := \{s, s + 1, \ldots, s + r - 1\}$ (for all $\mathbf{x} \in C$, $s \in [1, n - r + 1]$)

$$\mathbf{x} = (x_1, x_2, \ldots, x_{s-1}, x_s, x_{s+1}, x_{s+2}, \ldots, x_{s+r-1}, x_{s+r}, \ldots, x_n)$$
$$\mathbf{y} = (y_1, y_2, \ldots, y_{s-1}, y_s, y_{s+1}, \ldots, y_{s+r-2}, y_{s+r-1}, \ldots, y_{n-1})$$

Example: 2-bounded single deletion correcting code $n = 3$

$$C = \{000, 110, 011\}$$

<table>
<thead>
<tr>
<th>deletion position</th>
<th>${1,2}$</th>
<th>${2,3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>${00}$</td>
<td>${00}$</td>
</tr>
<tr>
<td>110</td>
<td>${10}$</td>
<td>${10, 11}$</td>
</tr>
<tr>
<td>011</td>
<td>${11, 01}$</td>
<td>${01}$</td>
</tr>
</tbody>
</table>
Definition: \( r \)-bounded single insertion correcting code

There exists a decoder which corrects a single insertion from the received sequence \( y \) and the range of insertion positions \([s, s + r]\) (for all \( x \in C, s \in [1, n - r + 1] \))

\[
\begin{align*}
x &= (x_1, x_2, \ldots, x_{s-1}, x_s, x_{s+1}, \ldots, x_{s+r-1}, x_{s+r}, \ldots, x_n) \\
y &= (y_1, y_2, \ldots, y_{s-1}, y_s, y_{s+1}, y_{s+2}, \ldots, y_{s+r}, y_{s+r+1}, \ldots, y_{n+1})
\end{align*}
\]

Example: 2-bounded single insertion correcting code \( n = 3 \)

\[
C = \{000, 110, 011\}
\]

<table>
<thead>
<tr>
<th>insertion position</th>
<th>{1,2,3}</th>
<th>{2,3,4}</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>{0000, 1000, 0100, 0010}</td>
<td>{0000, 0100, 0010, 0001}</td>
</tr>
<tr>
<td>110</td>
<td>{0110, 1010, 1100, 1110}</td>
<td>{1010, 1100, 1110, 1101}</td>
</tr>
<tr>
<td>011</td>
<td>{0011, 0101, 1011, 0111}</td>
<td>{0011, 0101, 0110, 0111}</td>
</tr>
</tbody>
</table>
Equivalence of insertion correction and deletion correction

Theorem 1

Code $C$ is an $r$-bounded single deletion correcting code

$\iff$ Code $C$ is an $r$-bounded single insertion correcting code

(c.f.)

Code $C$ is a single deletion correcting code

$\iff$ Code $C$ is a single insertion correcting code

If we want to prove that $C$ is $r$-bounded single insertion/deletion correcting,
then we need to only prove that $C$ is $r$-bounded single deletion correcting.
## Existing codes

### Substitution-Transposition (ST) code [Abdel-Ghaffar1998]

A 2-BSIDC code (proved by [Cheng2014]) \((a \in \{0, 1, 2\})\)

\[
\text{ST}_a(n) = \{ \mathbf{x} = (x_1, x_2, \ldots, x_n) \in \{0, 1\}^n \mid 1x_1 + 2x_2 + 1x_3 + 2x_4 + 1x_5 + \cdots \equiv a \pmod{3} \},
\]

### Shifted VT code [Schoeny2017]

An \(r\)-BSIDC code \((a \in \{0, 1, \ldots, r - 1\}, b \in \{0, 1\})\)

\[
\text{SVT}_{a,b}(n, r) = \{ \mathbf{x} \in \{0, 1\}^n \mid \sum_{i=1}^{n} ix_i \equiv a \pmod{r}, \sum_{i=1}^{n} x_i \equiv b \pmod{2} \}
\]

(Example) \(\text{SVT}_{a,b}(n, 4)\)

\[
\begin{cases}
  x_1 + 2x_2 + 3x_3 + \cdots + 1x_5 + 2x_6 + 3x_7 + \cdots \equiv a \pmod{4} \\
  x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + \cdots \equiv b \pmod{2}
\end{cases}
\]
Remarks for existing codes

SVT codes are not generalization of ST codes

\[ \text{ST}_a(n) = \{ \mathbf{x} \in \{0, 1\}^n | x_1 + 2x_2 + x_3 + 2x_4 + \cdots \equiv a \pmod{3} \} \]
\[ \text{SVT}_{a,b}(n, 2) = \{ \mathbf{x} \in \{0, 1\}^n | x_1 + 0x_2 + x_3 + 0x_4 + \cdots \equiv a \pmod{2}, \\
\hspace{1cm} x_1 + x_2 + x_3 + x_4 + \cdots \equiv b \pmod{2} \} \]

For \( n = 3 \)

\[ \text{ST}_0(3) = \{000, 110, 011\} \quad \text{SVT}_{0,0}(3, 2) = \{000, 101\} \]
\[ \text{ST}_1(3) = \{100, 001, 111\} \quad \text{SVT}_{0,1}(3, 2) = \{010, 111\} \]
\[ \text{ST}_2(3) = \{010, 101\} \quad \text{SVT}_{1,0}(3, 2) = \{110, 011\} \]
\[ \text{SVT}_{1,1}(3, 2) = \{100, 001\} \]

|ST\(_0\)(3)| > |SVT\(_{a,b}\)(3, 2)|
Contributions of this work

Construct two $r$-bounded SIDC codes (efficient decodable, large cardinality)

Exponential coefficient (EC) code

$$E_a(n, r) := \{ \mathbf{x} \in \{0, 1\}^n \mid \sum_{i=1}^{n} 2^{i-1} x_i \equiv a \pmod{2^{r-1} + 1} \}$$

- Generalization of ST codes
- Largest cardinality for $r \leq 3$

Odd coefficient (OC) code

$$O_a(n, r) := \{ \mathbf{x} \in \{0, 1\}^n \mid \sum_{i=1}^{n} (2i - 1) x_i \equiv a \pmod{2r} \} = \{ \mathbf{x} \mid x_1 + 3x_2 + 5x_3 + 7x_4 + \cdots \equiv a \pmod{2r} \}$$

- Largest cardinality for $r \geq 4$
Exponential coefficient code (1: Remarks)

\[ E_a(n, r) := \{ \mathbf{x} \in \{0, 1\}^n | \sum_{i=1}^{n} 2^{i-1} x_i \equiv a \pmod{2^{r-1} + 1} \} \]

\[ = \{ \mathbf{x} | x_1 + 2x_2 + 4x_3 + 8x_4 + \cdots \equiv a \pmod{2^{r-1} + 1} \} \]

(Examples)

\[ E_a(n, 2) = \{ \mathbf{x} | x_1 + 2x_2 + x_3 + 2x_4 + x_5 + 2x_6 + \cdots \equiv a \pmod{3} \}, \]
\[ E_a(n, 3) = \{ \mathbf{x} | x_1 + 2x_2 + 4x_3 + 3x_4 + x_5 + 2x_6 + \cdots \equiv a \pmod{5} \}. \]

Remark 1

EC codes \( E_a(n, r) \) are generalization of ST codes \((E_a(n, 2) = ST_a(n))\)

\[ ST_a(n) = \{ \mathbf{x} \in \{0, 1\}^n | x_1 + 2x_2 + x_3 + 2x_4 + x_5 + 2x_6 + \cdots \equiv a \pmod{3} \} \]
Proposition 2: Cardinality of EC code

For all $a, n, r$, the following holds:

$$|E_a(n, r)| = \begin{cases} 
\lceil \frac{2^n}{(2^r - 1) + 1} \rceil & \text{if } a < \text{rem}(2^n, 2^r - 1 + 1), \\
\lfloor \frac{2^n}{(2^r - 1) + 1} \rfloor & \text{if } a \geq \text{rem}(2^n, 2^r - 1 + 1).
\end{cases}$$

In particular, the maximum value achieves at $a = 0$

- $\lceil x \rceil$: ceiling function
- $\lfloor x \rfloor$: floor function
- $\text{rem}(A, B)$: remainder of $A \div B$
Exponential coefficient code (3: Decoding algorithm 1)

**Input:** Received word \( y = (y_1, y_2, \ldots, y_{n-1}) \), deletion range \([s, s + r - 1]\)

**Output:** Estimated word \( \hat{x} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) \in E_a(n, r) \)

Since the decoder knows deletion range,

\[
\hat{x}[1,s-1] = y[1,s-1], \quad \hat{x}[s+r,n] = y[s+r-1,n-1]
\]

\[
\begin{align*}
(x_1, x_2, \ldots, x_{s-1}, x_s, x_{s+1}, \ldots, x_{s+r-1}, x_{s+r}, \ldots, x_n) & = (y_1, y_2, \ldots, y_{s-1}, y_s, \ldots, y_{s+r-2}, y_{s+r-1}, \ldots, y_{n-1}) \text{ or } \\
& = (y_1, y_2, \ldots, y_{s-1}, y_s, y_{s+1}, \ldots, \ y_{s+r-1}, \ldots, y_{n-1}) \text{ or..} \\
& = (y_1, y_2, \ldots, y_{s-1}, y_s, y_{s+1}, \ldots, d, y_{s+r-1}, \ldots, y_{n-1})
\end{align*}
\]

Hence, we consider the decoding algorithm for the following code

\[
E_{b,s}(r) = \{(x_s, x_{s+1}, \ldots, x_{s+r-1}) \mid \sum_{i=s}^{s+r-1} 2^{i-1} x_i \equiv b \pmod{2^{r-1} + 1}\}
\]
Exponential coefficient code (4: Decoding algorithm 2)

\[ E_{b,s}(r) = \{(x_s, x_{s+1}, \ldots, x_{s+r-1}) \mid \sum_{i=s}^{s+r-1} 2^{i-1} x_i \equiv b \pmod{2^{r-1} + 1}\} \]
\[ = \{(x_1, x_2, \ldots, x_r) \mid \sum_{i=1}^{r-1} 2^{i-1} x_i \equiv 2^{-s+1} b \pmod{2^{r-1} + 1}\} \]
\[ = E_{2^{-s+1} b, 1}(r) \]

Thus, we consider the decoding algorithm for the following code:

\[ E_{a,1}(r) = \{(x_1, x_2, \ldots, x_r) \mid \sum_{i=1}^{r-1} 2^{i-1} x_i \equiv a \pmod{2^{r-1} + 1}\} \]
\[ = E_a(r, r) \]

Since \(|E_a(r, r)| \leq 2\) and 
\[ E_a(r, r) = \{\text{binary number for } a, \text{ binary number for } a + 2^{r-1} + 1\}, \]
the decoder calculate the Levenshtein distance between \(y\) and those codewords.
Odd coefficient code (1: Remarks)

\[
O_a(n, r) := \{ \mathbf{x} \in \{0, 1\}^n \mid \sum_{i=1}^{n} (2i - 1)x_i \equiv a \pmod{2r} \}
\]

\[
= \{ \mathbf{x} \mid x_1 + 3x_2 + 5x_3 + 7x_4 + \cdots \equiv a \pmod{2r} \}
\]

(Examples)

\[
O_a(n, 3) := \{ \mathbf{x} \mid x_1 + 3x_2 + 5x_3 + x_4 + 3x_5 + 5x_6 + \cdots \equiv a \pmod{6} \}
\]

\[
O_a(n, 4) := \{ \mathbf{x} \mid x_1 + 3x_2 + 5x_3 + 7x_4 + x_5 + 3x_6 + \cdots \equiv a \pmod{8} \}
\]

(Property)

\[
\sum_{i=1}^{n} (2i - 1)x_i \equiv a \pmod{2r} \Rightarrow \sum_{i=1}^{n} x_i \equiv a \pmod{2}
\]

Decoding algorithm for OC codes is similar one for SVT codes

(c.f.) SVT code

\[
SVT_{a,b}(n, r) = \{ \mathbf{x} \in \{0, 1\}^n \mid \sum_{i=1}^{n} ix_i \equiv a \pmod{r},
\quad \sum_{i=1}^{n} x_i \equiv b \pmod{2} \}
\]
Odd coefficient code (2: Cardinality 1)

Proposition 2: cardinality of OC code

Assume $n = rs + k$ (s is quotient of $n \div r$, k is reminder of $n \div r$)

$$|O_a(n, r)| = \frac{1}{2r} \left[ \sum_{d|r, d: odd} 2^{sr/d} \sum_{y \in \{0,1\}^k} c_d(a - \langle m_2, y \rangle) \right]$$

$$+ \sum_{d|r, d: even} 2^{sr/d} \sum_{y \in \{0,1\}^k} c_{2d}(a - \langle m_2, y \rangle)$$

$$\langle m_2, y \rangle := \sum_{i=1}^{k} (2i - 1)y_i$$

$$c_d(a) := \phi(d) \frac{\mu(d/(a, d))}{\phi(d/(a, d))}$$

$\phi(d)$: Euler's totient function, $\mu(d)$: Möbius function,
$(a, d)$: maximum common divisor of $a$ and $d$
Proposition 3: Cardinality of SVT code

Assume \( n = rs + k \) (\( s \) is quotient of \( n \div r \), \( k \) is reminder of \( n \div r \))

\[
|SVT_{a,b}(n,r)| = \frac{1}{2r} \sum_{d \mid r, d: \text{odd}} 2^{sr/d} \sum_{y \in \{0,1\}^k} c_d(a - \langle m_1, y \rangle)
\]

\( \langle m_1, y \rangle := \sum_{i=1}^{k} iy_i \)

Proposition 2 and 3 are derived from [Bibak2018]
Odd coefficient code (4: Cardinality 3)

In particular, for $n = rs$

$$|O_a(n, r)| = \frac{1}{2r} \sum_{d|r, d: \text{odd}} c_d(a)2^{n/d} + \frac{1}{2r} \sum_{d|r, d: \text{even}} c_{2d}(a)2^{n/d}$$

$$|\text{SVT}_{a,b}(n, r)| = \frac{1}{2r} \sum_{d|r, d: \text{odd}} c_d(a)2^{n/d}$$

The maximum achieves at $a = 0$:

$$|O_0(n, r)| = \frac{1}{2r} \sum_{d|r, d: \text{odd}} c_d(0)2^{n/d} + \frac{1}{2r} \sum_{d|r, d: \text{even}} c_{2d}(0)2^{n/d}$$

$$|\text{SVT}_{0,b}(n, r)| = \frac{1}{2r} \sum_{d|r, d: \text{odd}} c_d(0)2^{n/d}$$

If $r$ is even, $\max |O_a(n, r)| > \max |\text{SVT}_{a,b}(n, r)|$

If $r$ is odd, $\max |O_a(n, r)| = \max |\text{SVT}_{a,b}(n, r)|$
Odd coefficient code (5: Decoding algorithm 1)

**Input:** Received word \( y = (y_1, y_2, \ldots, y_{n-1}) \), Deletion range \([s, s + r - 1]\)

**Output:** Estimate word \( \hat{x} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) \in O_a(n, r) \)

Since the decoder knows deletion range,

\[
\hat{x}[1,s-1] = y[1,s-1], \quad \hat{x}[s+r,n] = y[s+r-1,n-1]
\]

\[
(x_1, x_2, \ldots, x_{s-1}, x_s, x_{s+1}, \ldots, x_{s+r-1}, x_{s+r}, \ldots, x_n)
\]

\[
= (y_1, y_2, \ldots, y_{s-1}, d, y_s, \ldots, y_{s+r-2}, y_{s+r-1}, \ldots, y_{n-1}) \text{ or }
\]

\[
= (y_1, y_2, \ldots, y_{s-1}, y_s, d, \ldots, y_{s+r-2}, y_{s+r-1}, \ldots, y_{n-1}) \text{ or...}
\]

\[
= (y_1, y_2, \ldots, y_{s-1}, y_s, y_{s+1}, \ldots, d, y_{s+r-1}, \ldots, y_{n-1})
\]

Hence, we consider the decoding algorithm for the following code

\[
O_{b,s}(r) = \{(x_s, x_{s+1}, \ldots, x_{s+r-1}) \mid \sum_{i=s}^{s+r-1} (2i - 1)x_i \equiv b \pmod{2r}\}\
\]
Odd coefficient code (6: Decoding algorithm 2)

\[ O_{b,s}(r) = \{(x_s, x_{s+1}, \ldots, x_{s+r-1}) \mid \sum_{i=s}^{s+r-1} (2i - 1)x_i \equiv b \pmod{2r}\} \]

Similar to the decoding algorithm for SVT code

**Require:** Received sequence \( y \), code parameters \( (a, k, r) \)

**Ensure:** Estimated sequence \( \hat{x} \)

1. Calculate \( w = h_w(y) \) and \( b = \sum_{i=1}^{r-1} (2i + 2k - 3)y_i \)
2. Set \( \lambda = \text{rem}(a - w, 2) \)
3. **if** \( \lambda = 0 \) **then**
4. Calculate \( R_1 = \text{rem}(a - b, 2r)/2 \)
5. Search \( p \) such that \( h_w(y_{[p,r-1]}) = R_1 \)
6. Output \( \hat{x} = y_{\leftarrow (p,0)} \)
7. **else**
8. Calculate \( L_0 = \text{rem}(a - b + 1 - 2(k + w), 2r)/2 \)
9. Search \( p \) such that \( \{|i \in [1, p] \mid y_i = 0\}| = L_0 \)
10. Output \( \hat{x} = y_{\leftarrow (p+1,1)} \)
11. **end if**
## Comparison of cardinalities

<table>
<thead>
<tr>
<th>$r$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\max_{a,b}</td>
<td>\text{SVT}_{a,b}(10, r)</td>
<td>$</td>
<td>256</td>
<td>172</td>
<td>128</td>
</tr>
<tr>
<td>$\max_a</td>
<td>E_a(10, r)</td>
<td>$</td>
<td>342</td>
<td>205</td>
<td>114</td>
</tr>
<tr>
<td>$\max_a</td>
<td>O_a(10, r)</td>
<td>$</td>
<td>272</td>
<td>172</td>
<td>136</td>
</tr>
<tr>
<td>$\max_{a,b}</td>
<td>\text{SVT}_{a,b}(11, r)</td>
<td>$</td>
<td>512</td>
<td>344</td>
<td>256</td>
</tr>
<tr>
<td>$\max_a</td>
<td>E_a(11, r)</td>
<td>$</td>
<td>683</td>
<td>410</td>
<td>228</td>
</tr>
<tr>
<td>$\max_a</td>
<td>O_a(11, r)</td>
<td>$</td>
<td>528</td>
<td>344</td>
<td>266</td>
</tr>
</tbody>
</table>

- EC codes have largest cardinalities for $r \leq 3$
- OC codes have largest cardinalities for $r \geq 4$
  - If $r$ is even, $\max | \text{SVT}_{a,b}(n, r)| < \max | O_a(n, r)|$
  - If $r$ is odd, $\max | \text{SVT}_{a,b}(n, r)| = \max | O_a(n, r)|$
Conclusion

- Construct $r$-bounded SIDC codes with larger cardinalities
  - Exponential coefficient (EC) code
  - Odd coefficient (OC) code
- Evaluate the cardinalities
  - If $r \leq 3$, EC codes have largest cardinalities
  - If $r \geq 4$, OC codes have largest cardinalities

(Construction 1) Exponential coefficient code

\[ E_a(n, r) := \{ \mathbf{x} \in \{0, 1\}^n \mid \sum_{i=1}^{n} 2^{i-1}x_i \equiv a \pmod{2^{r-1} + 1} \} \]

(Construction 2) Odd coefficient code

\[ O_a(n, r) := \{ \mathbf{x} \in \{0, 1\}^n \mid \sum_{i=1}^{n} (2i - 1)x_i \equiv a \pmod{2r} \} \]