Guess & Check Codes for Deletions and Synchronization

Serge Kas Hanna

Joint work with
Salim El Rouayheb

Illinois Institute of Technology
Motivation

• Deletions: 10101010 → 110010

• Deletions were first studied by Varshamov-Tenengolts ('65) and Levenshtein ('66)

• Our motivation: file synchronization, E.g. Dropbox

• Recent application: DNA-based storage
Deletion channel

- Capacity of the deletion channel, still an open problem
  - Probabilistic model: deletions are iid and occur with probability $p$
  - Lower and upper bounds: [Mitzenmacher and Drinea ’06], [Diggavi et al. ’07], [Kanoria and Montanari ’13], [Venkataramanan et al. ’13], [Rahmati and Duman ’15]

- Codes for correcting a fixed number of deletions
  - Single deletion: Varshamov-Tenengluts (’65), VT syndrome:
    \[ \sum_{i=1}^{n} ix_i \pmod{n+1} \]
  - Two or more deletions: open problem for more than 50 years to construct codes with asymptotically optimal redundancy
Problem and Contributions

- **Problem**
  - Correct fixed number of $\delta > 1$ deletions in binary strings
  - Design polynomial time encoding and decoding schemes
  - Asymptotically optimal redundancy

- **Fundamental limits:** [Levenshtein ’66], [Cullina and Kiyavash ’14]
  - Asymptotic number of redundant bits needed: $c\delta \log n$, for some $c > 0$

- **Previous work:** zero-error decoding
  - Our approach: asymptotically vanishing probability of decoding failure, assuming uniform iid message

- **Contributions:** new family of explicit codes called **Guess & Check Codes**
## Previous Work

<table>
<thead>
<tr>
<th>Number of Deletions</th>
<th>Redundancy</th>
<th>Polynomial time encoding &amp; decoding</th>
<th>Error</th>
<th>Scheme applies for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Varshamov-Tenengolts ('65)</td>
<td>One</td>
<td>$\log(n + 1)$</td>
<td>✔</td>
<td>Zero-error</td>
</tr>
<tr>
<td>Schulman and Zuckerman ('99)</td>
<td>$\alpha \cdot n$  $\alpha &lt; 1$</td>
<td>$O(n)$</td>
<td>✔</td>
<td>Zero-error</td>
</tr>
<tr>
<td>Helberg and Ferreira ('02)</td>
<td>$\delta &gt; 1$</td>
<td>$O(n)$</td>
<td>❌</td>
<td>Zero-error</td>
</tr>
<tr>
<td>Brankensiek et al. ('16)</td>
<td>$\delta &gt; 1$</td>
<td>$O(\delta^2 \log \delta \log n)$</td>
<td>✔</td>
<td>Zero-error</td>
</tr>
<tr>
<td><strong>Guess &amp; Check codes</strong></td>
<td>$\delta &gt; 1$</td>
<td>$c\delta \log k$  $c &gt; \delta$</td>
<td>✔</td>
<td><strong>Asymptotically vanishing</strong></td>
</tr>
</tbody>
</table>

- **Varshamov-Tenengolts ('65)**: One deletion, $\log(n + 1)$ redundancy, polynomial time encoding & decoding, zero-error for arbitrary strings.
- **Schulman and Zuckerman ('99)**: $\alpha \cdot n$, $\alpha < 1$, $O(n)$ redundancy, polynomial time encoding & decoding, zero-error for arbitrary strings.
- **Helberg and Ferreira ('02)**: $\delta > 1$, $O(n)$ redundancy, not polynomial time encoding & decoding, zero-error for arbitrary strings.
- **Brankensiek et al. ('16)**: $\delta > 1$, $O(\delta^2 \log \delta \log n)$ redundancy, polynomial time encoding & decoding, zero-error for “pattern rich” strings.

**Guess & Check codes** have $\delta > 1$, $c\delta \log k$, $c > \delta$ redundancy, polynomial time encoding & decoding, asymptotically vanishing error for arbitrary strings.
GC Code Example for 1 Deletion

- **Encoding the 16-bit message:**
  - GF(17)
  - \[1110\ 0000\ 1101\ 0001\]
  - 14 0 2 1
  - (6,4) MDS
  - 2 parities

- **Assume WLOG deletion affects systematic bits, and 14th bit gets deleted**
  - 1110000011010001
  - 111000001101001

- **Decoding**
  - \(\text{Guess 1: } 1110000011010001\)
    - Decoded using 1st parity
    - 1 0 6 10
    - Check with 2nd parity
    - 13
  - \(\text{Guess 2: } 1110000011010001\)
    - 14 13 6 10
    - 13
  - \(\text{Guess 3: } 1110000011010001\)
    - 14 0 7 10
    - 13
  - \(\text{Guess 4: } 1110000011010001\)
    - 14 0 2 1
    - Success
Simulations – Probability of Decoding Failure of GC codes for 1 Deletion.
Simulations – 3 Deletions

![Probability of Decoding Failure of GC Codes for 3 Deletions](image)

**Average running time:**

order of milliseconds
Theorem 1: Guess & Check (GC) codes can correct in polynomial time up to a constant number of $\delta$ deletions occurring in a binary string. Let $c > \delta$ be a constant integer. The code has the following properties:

- Asymptotically optimal redundancy: $c(\delta + 1) \log k$
- Polynomial encoding complexity: $O(k / \log k)$
- Polynomial decoding complexity: $O(k^{\delta+2} / \log^\delta k)$
- Vanishing probability of decoding failure: $O(k^{2\delta-c} / \log^\delta k)$
What Causes a Decoding Failure?

- Decoding failure: more than one possible guess, different decoded strings

- Example: 16-bit message 0000100011110110

  - (6,4) MDS encoding over GF(17): 0 8 15 6 12 5 2 parities

  - Suppose 14th bit gets deleted, decoding:
    - Guess 1: 8 4 7 10 ✔
    - Guess 2: 8 4 7 10 ❌
    - Guess 3: 8 4 7 10 ❌
    - Guess 4: 0 8 15 6 ✔

    Guesses 1 & 4 satisfy the 2 parities

- Probability of decoding failure for a given string: combinatorial problem that depends on the string and deletion position

- Proof approach: assume message is uniform iid, average over all possible messages
Decoding Failure – 1 Deletion

Assume WLOG that Guess 1 is correct, observe the output of decoder at wrong Guess $i \neq 1$

Set of all transmitted k-bit strings

Set of all GC decoder outputs

Set satisfying first parity

Set satisfying all $c$ parities

$Pr(\text{decoding failure in guess } i \neq 1) = Pr(\text{decoded string is in } A)$

**Lemma**: at most 2 different transmitted sequences can lead to the same decoded string in any Guess $i \neq 1$
Proof of $\Pr(F)$ for One Deletion

$Pr(F) \leq Pr \left( \bigcup_{i=2}^{k/\log k} \{ \mathcal{Y}_i \in A, \mathcal{Y}_i \neq \mathcal{Y}_1 \} \right)$  \hspace{1cm} (1)

Union bound

$\leq \sum_{i=2}^{k/\log k} Pr(\mathcal{Y}_i \in A, \mathcal{Y}_i \neq \mathcal{Y}_1)$  \hspace{1cm} (2)

$\leq \sum_{i=2}^{k/\log k} Pr(\mathcal{Y}_i \in A)$  \hspace{1cm} (3)

Lemma

$= \sum_{i=2}^{k/\log k} \sum_{Y \in A} Pr(\mathcal{Y}_i = Y)$  \hspace{1cm} (4)

$\leq \sum_{i=2}^{k/\log k} \sum_{Y \in A} \frac{2}{|B|}$  \hspace{1cm} (5)

Subspace cardinality

$= 2 \left( \frac{k}{\log k} - 1 \right) \frac{|A|}{|B|}$  \hspace{1cm} (6)

$= 2 \left( \frac{k}{\log k} - 1 \right) \frac{q^{k/\log k-c}}{q^{k/\log k-1}}$  \hspace{1cm} (7)

$< \frac{k^{c-2}}{\log k}$.  \hspace{1cm} (8)

$k$ : length of message

$\mathcal{Y}_i$ : string decoded in Guess $i$

$c$ : number of parities

$A$ : set satisfying all $c$ parities

$B$ : set satisfying first parity

$q$ : field size
Interactive synchronization algorithm by [Venkataramanan et al. ’15]
- Isolate single deletions, use VT codes
- Modification: isolate $\delta$ or fewer deletions, use GC codes

Gain: (1) less communication rounds, (2) lower communication cost

Up to 75% improvement
Up to 15% improvement
Summary

• Guess & Check Codes
  - Correct multiple deletions
  - Explicit codes with asymptotically optimal redundancy
  - Deterministic polynomial time encoding and decoding schemes
  - Asymptotically vanishing probability of decoding failure

• Redundancy-Complexity-Probability of failure trade-offs

• GC codes can also correct insertions

• GC codes can be used to efficiently correct localized deletions/insertions (e.g. bursts)