Coding against deletions in oblivious and online models

Ray Li (CMU, rayyli@stanford.edu)*
Venkatesan Guruswami (CMU, venkatg@cs.cmu.edu)

Carnegie Mellon University

SODA 2018
New Orleans, LA, January 7 2018
Outline for section 1

1 Motivating problem
   - Combinatorics formulation

2 Adversarial deletions

3 Online deletions

4 Oblivious deletions

5 Conclusion
This problem is trivial

Longest common subsequence

\[ w_1 = 0101010 \]
\[ w_2 = 0000111 \]

\[ |w_1| = |w_2| = 7 \]

\[ \text{LCS}(w_1, w_2) = 4 \]
This problem is trivial

*Longest common subsequence*

\[ w_1 = 0101010 \]
\[ w_2 = 0000111 \]

\[ |w_1| = |w_2| = 7 \]

\[ \text{LCS}(w_1, w_2) = 4 \]
This problem is trivial

*Longest common subsequence:*

\[ w_1 = 0101010 \]
\[ w_2 = 0000111 \]

**Question:** What is the maximum number of length \( N \) strings such that, for any two \( w_1, w_2 \), we have \( \text{LCS}(w_1, w_2) < 0.5N \)?
This problem is trivial

Longest common subsequence:

\[ w_1 = 0101010 \]
\[ w_2 = 0000111 \]

Question: What is the maximum number of length \( N \) strings such that, for any two \( w_1, w_2 \), we have \( \text{LCS}(w_1, w_2) < 0.5N \)? Answer: 2

\[
\begin{align*}
0000000 \\
1111111
\end{align*}
\]
This problem is trivial

Longest common subsequence:

\[ w_1 = 0101010 \]
\[ w_2 = 0000111 \]

Question: What is the maximum number of length \( N \) strings such that, for any two \( w_1, w_2 \), we have \( \text{LCS}(w_1, w_2) < 0.5N \)? Answer: 2

1111000 Majority: 1
0000111 Majority: 0
0101010 Majority: 0
This problem is trivial

*Longest common subsequence:*

\[ w_1 = 0101010 \]
\[ w_2 = 0000111 \]

Question: What is the maximum number of length \( N \) strings such that, for any two \( w_1, w_2 \), we have \( \text{LCS}(w_1, w_2) < 0.5N \)? Answer: 2


\[ \begin{array}{cc}
0000111 & \text{Majority: 0} \\
0101010 & \text{Majority: 0}
\end{array} \]
This problem is Longest common subsequence:

\[ w_1 = 0101010 \]
\[ w_2 = 0000111 \]

Question: What is the maximum number of length \( N \) strings such that, for any two \( w_1, w_2 \), we have \( \text{LCS}(w_1, w_2) < 0.501N \)?

\( O(1)? \quad \alpha \log N? \quad \text{poly } N? \quad 2^{\alpha N}? \)
Outline for section 2

1 Motivating problem

2 Adversarial deletions
   - What are error correcting codes?
   - Motivating problem, take 2

3 Online deletions

4 Oblivious deletions

5 Conclusion
Error correcting codes allow noisy channel communication

\[ m = 01 \rightarrow \text{Alice} \]
Error correcting codes allow noisy channel communication

$m = 01 \rightarrow \text{Alice} \quad \text{Bob}$
Error correcting codes allow noisy channel communication

\[ m = 01 \rightarrow \text{Alice} \quad \text{Channel} \quad \text{Bob} \]
Error correcting codes allow noisy channel communication.

$m = 01 \rightarrow$ Alice $\rightarrow_{01}$ Channel $\rightarrow_{1}$ Bob
Error correcting codes allow noisy channel communication

\[ m = 01 \rightarrow \text{Alice} \rightarrow 01 \rightarrow \text{Channel} \rightarrow 1 \rightarrow \text{Bob} \rightarrow m = 10? \]
Error correcting codes allow noisy channel communication

We consider channels applying constant fraction of deletions

<table>
<thead>
<tr>
<th>Error type</th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substitution</td>
<td>000</td>
<td>001</td>
</tr>
<tr>
<td>Erasure</td>
<td>000</td>
<td>00⊥</td>
</tr>
<tr>
<td>Deletion</td>
<td>000</td>
<td>00</td>
</tr>
</tbody>
</table>

1. First studied by Levenshtein 1960s (synchronization errors)
2. Deletions poorly understood compared to substitutions and erasures
Error correcting codes allow noisy channel communication

\[ m = 01 \]

\[ \text{Alice (Enc)} \]

\[ \text{Bob (Dec)} \]
Error correcting codes allow noisy channel communication.
Error correcting codes allow noisy channel communication

\[ m = 01 \]

Alice (Enc) \[ \rightarrow \] Channel \[ \rightarrow \] Bob (Dec)

Channel input: 000111

Bob input: 00011
Error correcting codes allow noisy channel communication

\[ m = 01 \rightarrow \text{Alice (Enc)} \rightarrow 000111 \rightarrow \text{Channel} \rightarrow 00011 \rightarrow \text{Bob (Dec)} \rightarrow m = 01 \]
Error correcting codes allow noisy channel communication

Two questions

1. Tradeoff between “redundancy” (rate) and “robustness” (error fraction) of the code (i.e. channel capacity). This work.

2. Efficient construction, encoding, decoding.
Error correcting codes allow noisy channel communication

- **Message set** $\mathcal{M} = \{00, 01, 10, 11\}$
- **Code** $C = \{000000, 010101, 101010, 111111\}$ of **codewords**
- **Alphabet** is binary (for the rest of the talk)
- **Length (blocklength)** is $N = 6$
- **Rate** $R = \frac{\log |\mathcal{M}|}{N} = \frac{1}{3}$
- **Encoder** $\text{Enc} : \mathcal{M} \rightarrow C$, $\text{Enc} : m \mapsto m m m$
- **Decoder** $\text{Dec} : \Sigma^* \rightarrow C$, $\text{Dec} : x_1 x_2 \ldots x_5 \mapsto x_1 x_5$
Error correcting codes allow noisy channel communication

\[ m = 01 \rightarrow \text{Alice (Enc)} \rightarrow \text{Channel} \rightarrow \text{Bob (Dec)} \rightarrow m = 01 \]

**Rate**
- \( R = \frac{\log |\mathcal{M}|}{N} \in (0, 1) \).
- \( R \) is proportion of non-redundant symbols
- Want family of constant rate codes (implicitly \( N \rightarrow \infty \)), a.k.a. rate bounded from 0.
- “constant rate” means “\(|\mathcal{M}| = 2^{\Omega(N)}\)”
Adversarial deletions is equivalent to LCS question

\[
LCS(C) = \max_{c \neq c' \in C} LCS(c, c')
\]

**Deletion:** $000 \leftrightarrow 00.$

**Adversarial:** Dec *always* returns the correct $m$ when Noise is at most $t$ del.

**Lemma (Levenshtein 66)**

The following are equivalent.

- $LCS(C) \leq N - t - 1.$
- $C$ decodable under up to $t$ adversarial deletions.
Adversarial deletions is equivalent to LCS question

\[ \text{LCS}(C) = \max_{c \neq c' \in C} \text{LCS}(c, c') \]

**Lemma (Levenshtein 66)**

*The following are equivalent.*

- \( \text{LCS}(C) \leq N - t - 1 \).
- \( C \) decodable under up to \( t \) adversarial deletions.

**Q:** \( \exists |C| > 2^{\Omega(N)} \) strings with \( \text{LCS}(C) < 0.501N \)?
Adversarial deletions is equivalent to LCS question

\[ \text{LCS}(C) = \max_{c \neq c' \in C} \text{LCS}(c, c') \]

**Lemma (Levenshtein 66)**

*The following are equivalent.*

- \( \text{LCS}(C) \leq N - t - 1. \)
- \( C \) decodable under up to \( t \) adversarial deletions.

**Q:** \( \exists |C| > 2^{\Omega(N)} \) strings with \( \text{LCS}(C) < 0.501N \)?

**Q v2:** \( \exists \) constant rate \( C \) for \( 0.499N \) adversarial deletions?
Adversarial deletions is equivalent to LCS question

$LCS(C) = \max_{c \neq c' \in C} LCS(c, c')$

**Lemma (Levenshtein 66)**

_The following are equivalent._

- $LCS(C) \leq N - t - 1$.
- $C$ decodable under up to $t$ adversarial deletions.

**Q:** $\exists |C| > 2^{\Omega(N)}$ strings with $LCS(C) < 0.501N$?

**Q v2:** $\exists$ constant rate $C$ for $0.499N$ adversarial deletions? **A:** Open.
Adversarial deletions is equivalent to LCS question

Want: *Capacity* (Optimal rate vs. *p* tradeoff) of adversarial deletions.

Zero rate threshold is smallest *p* where capacity is 0.

**Definition (Zero Rate Threshold)**

The zero rate threshold of adversarial deletions, *p*_{0}^{(adv)}, is largest *p* where constant rate codes correct *pN* adversarial deletions.
Adversarial deletions is equivalent to LCS question

**Definition (Zero Rate Threshold)**

The zero rate threshold of adversarial deletions, $p_{0}^{(adv)}$, is largest $p$ where constant rate codes correct $pN$ adversarial deletions.

**Q:** $\exists$ $C$ for $pN$ adversarial deletions for $p = 0.499$? **A:** Open.

**Q:** What is $p_{0}^{(adv)}$?
Adversarial deletions is equivalent to LCS question

**Definition (Zero Rate Threshold)**

The zero rate threshold of adversarial deletions, $p_{0}^{(\text{adv})}$, is largest $p$ where constant rate codes correct $pN$ adversarial deletions.

**Q:** $\exists$ $C$ for $pN$ adversarial deletions for $p = 0.499$? **A:** Open.

**Q:** What is $p_{0}^{(\text{adv})}$?

- $p_{0}^{(\text{adv})} \leq 1/2$ (Trivial)
Adversarial deletions is equivalent to LCS question

**Definition (Zero Rate Threshold)**

The zero rate threshold of adversarial deletions, $p_0^{(\text{adv})}$, is largest $p$ where constant rate codes correct $pN$ adversarial deletions.

**Q:** ∃ $C$ for $pN$ adversarial deletions for $p = 0.499$? **A:** Open.

**Q:** What is $p_0^{(\text{adv})}$?

- $p_0^{(\text{adv})} \leq 1/2$ (Trivial)
- $p_0^{(\text{adv})} \geq H^{-1}(0.5) \approx 0.11$ (Greedy)
Adversarial deletions is equivalent to LCS question

Definition (Zero Rate Threshold)

The zero rate threshold of adversarial deletions, $p_0^{(adv)}$, is largest $p$ where constant rate codes correct $pN$ adversarial deletions.

Q: $\exists \ C$ for $pN$ adversarial deletions for $p = 0.499$? A: Open.

Q: What is $p_0^{(adv)}$?

- $p_0^{(adv)} \leq 1/2$ (Trivial)
- $p_0^{(adv)} \geq H^{-1}(0.5) \approx 0.11$ (Greedy)
- $p_0^{(adv)} \geq 0.17$ ([Kash, Mitzenmacher, Thaler, Ullman '11], Random codes)
Adversarial deletions is equivalent to LCS question

**Definition (Zero Rate Threshold)**

The **zero rate threshold of adversarial deletions**, $p_{0}^{(adv)}$, is largest $p$ where constant rate codes correct $pN$ adversarial deletions.

**Q:** ∃ $C$ for $pN$ adversarial deletions for $p = 0.499$? **A:** Open.

**Q:** What is $p_{0}^{(adv)}$?

- $p_{0}^{(adv)} \leq 1/2$ (Trivial)
- $p_{0}^{(adv)} \geq H^{-1}(0.5) \approx 0.11$ (Greedy)
- $p_{0}^{(adv)} \geq 0.17$ ([Kash, Mitzenmacher, Thaler, Ullman '11], Random codes)
- $p_{0}^{(adv)} \geq 1/3$ ([Bukh, Guruswami '16])

Closing the gap between $0.41$ and $0.5$ is open.
Adversarial deletions is equivalent to LCS question

**Definition (Zero Rate Threshold)**

The zero rate threshold of adversarial deletions, \( p_0^{(adv)} \), is largest \( p \) where constant rate codes correct \( pN \) adversarial deletions.

Q: \( \exists \ C \) for \( pN \) adversarial deletions for \( p = 0.499 \)? A: Open.

Q: What is \( p_0^{(adv)} \)?

- \( p_0^{(adv)} \leq 1/2 \) (Trivial)
- \( p_0^{(adv)} \geq H^{-1}(0.5) \approx 0.11 \) (Greedy)
- \( p_0^{(adv)} \geq 0.17 \) ([Kash, Mitzenmacher, Thaler, Ullman '11], Random codes)
- \( p_0^{(adv)} \geq 1/3 \) ([Bukh, Guruswami '16])
- \( p_0^{(adv)} \geq 0.41 \) ([Bukh, Guruswami, Håstad, 17])

Closing the gap between 0.41 and 0.5 is open.
Adversarial deletions is equivalent to LCS question

**Definition (Zero Rate Threshold)**

The zero rate threshold of adversarial deletions, $p_{0}^{\text{(adv)}}$, is largest $p$ where constant rate codes correct $pN$ adversarial deletions.

**Q:** $\exists C$ for $pN$ adversarial deletions for $p = 0.499$? **A:** Open.

**Q:** What is $p_{0}^{\text{(adv)}}$?

- $p_{0}^{\text{(adv)}} \leq 1/2$ (Trivial)
- $p_{0}^{\text{(adv)}} \geq H^{-1}(0.5) \approx 0.11$ (Greedy)
- $p_{0}^{\text{(adv)}} \geq 0.17$ ([Kash, Mitzenmacher, Thaler, Ullman '11], Random codes)
- $p_{0}^{\text{(adv)}} \geq 1/3$ ([Bukh, Guruswami '16])
- $p_{0}^{\text{(adv)}} \geq 0.41$ ([Bukh, Guruswami, Håstad, 17])

Closing the gap between 0.41 and 0.5 is open.
(Adversarial) Deletions are poorly understood

Want: *Capacity* (Optimal rate vs. \( p \) tradeoff) of adversarial deletions.

<table>
<thead>
<tr>
<th>Error type</th>
<th>( p^{(adv)}_0 )</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substitutions</td>
<td>( 1/4 )</td>
<td>?</td>
</tr>
<tr>
<td>Erasures</td>
<td>( 1/2 )</td>
<td>?</td>
</tr>
<tr>
<td>Deletions</td>
<td>( [0.41, 0.5] )</td>
<td>??</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error type</th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substitution</td>
<td>000</td>
<td>001</td>
</tr>
<tr>
<td>Erasure</td>
<td>000</td>
<td>00⊥</td>
</tr>
<tr>
<td>Deletion</td>
<td>000</td>
<td>00</td>
</tr>
</tbody>
</table>
This work considers online and oblivious deletion codes

Error Models
- Adversarial: Channel gets full knowledge of the codeword
- Random: i.i.d deletions
- **Online**: In deciding to corrupt $x_i$, channel knows $x_1 \ldots x_i$.
- **Oblivious**: Channel knows $M$, Enc, Dec, and transmitted message, but not codeword
Outline for section 3

1. Motivating problem
2. Adversarial deletions
3. Online deletions
4. Oblivious deletions
5. Conclusion
What is an online channel?

An online channel erases/flips/deletes the $i$th bit of a codeword $c$ only knowing the first $i$ bits, and corrupts $\leq pN$ bits.
\[ p_0^{(on)} \leq 1/2 \]

An online channel erases/flips/deletes the \( i \)th bit of a codeword \( c \) only knowing the first \( i \) bits, and corrupts \( \leq pN \) bits.

**Fact**

*There no nontrivial codes against an online deletion channel with \( p \geq 1/2 \).*

In other words, \( p_0^{(on)} \leq 1/2 \).

Proof: Pick random \( b \in \{0, 1\} \). Delete every \( b \).

\[ s \quad 0 \]
\( p_0^{(on)} \leq 1/2 \)

An online channel erases/flips/deletes the \( i \)th bit of a codeword \( c \) only knowing the first \( i \) bits, and corrupts \( \leq pN \) bits.

**Fact**

*There no nontrivial codes against an online deletion channel with \( p \geq 1/2 \). In other words, \( p_0^{(on)} \leq 1/2 \).*

Proof: Pick random \( b \in \{0, 1\} \). Delete every \( b \).

\[
s \begin{array}{c}
0 \\
1 
\end{array}
\]
$p_0^{(on)} \leq 1/2$

An online channel erases/flips/deletes the $i$th bit of a codeword $c$ only knowing the first $i$ bits, and corrupts $\leq pN$ bits.

**Fact**

*There no nontrivial codes against an online deletion channel with $p \geq 1/2$. In other words, $p_0^{(on)} \leq 1/2$.*

Proof: Pick random $b \in \{0, 1\}$. Delete every $b$.

$$s \begin{array}{c} 0 \end{array}$$
$p_{0}^{(on)} \leq 1/2$

An online channel erases/flips/deletes the $i$th bit of a codeword $c$ only knowing the first $i$ bits, and corrupts $\leq pN$ bits.

**Fact**

*There no nontrivial codes against an online deletion channel with $p \geq \frac{1}{2}$. In other words, $p_{0}^{(on)} \leq \frac{1}{2}$.***

Proof: Pick random $b \in \{0, 1\}$. Delete every $b$.

$$s 0 0$$
$p_0^{(on)} \leq 1/2$

An online channel erases/flips/deletes the $i$th bit of a codeword $c$ only knowing the first $i$ bits, and corrupts $\leq pN$ bits.

Fact

There no nontrivial codes against an online deletion channel with $p \geq \frac{1}{2}$.

In other words, $p_0^{(on)} \leq \frac{1}{2}$.

Proof: Pick random $b \in \{0, 1\}$. Delete every $b$.

\[
\begin{array}{c}
\text{s} \\
000
\end{array}
\]
\[ p_0^{(\text{on})} \leq 1/2 \]

An online channel erases/flips/deletes the \( i \)th bit of a codeword \( c \) only knowing the first \( i \) bits, and corrupts \( \leq pN \) bits.

**Fact**

*There no nontrivial codes against an online deletion channel with \( p \geq \frac{1}{2} \).*

In other words, \( p_0^{(\text{on})} \leq \frac{1}{2} \).

Proof: Pick random \( b \in \{0, 1\} \). Delete every \( b \).

\[
\begin{array}{c}
0 \\
0 \\
0 \\
0 \\
\end{array}
\]
$p_{0}^{(\text{on})} \leq 1/2$

An online channel erases/flips/deletes the $i$th bit of a codeword $c$ only knowing the first $i$ bits, and corrupts $\leq pN$ bits.

Fact

There no nontrivial codes against an online deletion channel with $p \geq \frac{1}{2}$. In other words, $p_{0}^{(\text{on})} \leq \frac{1}{2}$.

Proof: Pick random $b \in \{0, 1\}$. Delete every $b$.

$$s \quad 00001$$
\( p_{0}^{(on)} \leq 1/2 \)

An online channel erases/flips/deletes the \( i \)th bit of a codeword \( c \) only knowing the first \( i \) bits, and corrupts \( \leq pN \) bits.

**Fact**

There no nontrivial codes against an online deletion channel with \( p \geq \frac{1}{2} \).

In other words, \( p_{0}^{(on)} \leq \frac{1}{2} \).

Proof: Pick random \( b \in \{0,1\} \). Delete every \( b \).

\[
\begin{array}{c}
s \\
0000
\end{array}
\]
$p_0^{\text{(on)}} \leq 1/2$

An online channel erases/flips/deletes the $i$th bit of a codeword $c$ only knowing the first $i$ bits, and corrupts $\leq pN$ bits.

**Fact**

*There no nontrivial codes against an online deletion channel with $p \geq \frac{1}{2}$. In other words, $p_0^{\text{(on)}} \leq \frac{1}{2}$.***

Proof: Pick random $b \in \{0, 1\}$. Delete every $b$.

$s \hspace{1cm} 00001$
$p_0^{(on)} \leq 1/2$

An online channel erases/flips/deletes the $i$th bit of a codeword $c$ only knowing the first $i$ bits, and corrupts $\leq pN$ bits.

**Fact**

*There no nontrivial codes against an online deletion channel with $p \geq \frac{1}{2}$. In other words, $p_0^{(on)} \leq \frac{1}{2}$.*

Proof: Pick random $b \in \{0, 1\}$. Delete every $b$.

\[ s \quad 0000 \]
$p_0^{(on)} \leq 1/2$

An online channel erases/flips/deletes the $i$th bit of a codeword $c$ only knowing the first $i$ bits, and corrupts $\leq pN$ bits.

**Fact**

*There no nontrivial codes against an online deletion channel with $p \geq 1/2$. In, other words, $p_0^{(on)} \leq 1/2$.***

Proof: Pick random $b \in \{0, 1\}$. Delete every $b$.

$$s \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$
(Online) Deletions are poorly understood

Recall: *Capacity* is optimal Rate vs. $p$ tradeoff

**Q.** What is the capacity of online deletions?

<table>
<thead>
<tr>
<th>Error type</th>
<th>$p_0^{(on)}$</th>
<th>Capacity</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substitutions</td>
<td>$1/4$</td>
<td>Known</td>
<td>[Dey, Jaggi, Langberg, Sarwate '12]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Chen, Jaggi, Langberg, '15]</td>
</tr>
<tr>
<td>Erasures</td>
<td>$1/2$</td>
<td>$1 - 2p$</td>
<td>[Bassily, Smith '14]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Chen, Jaggi, Langberg, '15]</td>
</tr>
<tr>
<td>Deletions</td>
<td>[0.41, 0.5]</td>
<td>??</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error type</th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substitution</td>
<td>000</td>
<td>001</td>
</tr>
<tr>
<td>Erasure</td>
<td>000</td>
<td>00⊥</td>
</tr>
<tr>
<td>Deletion</td>
<td>000</td>
<td>00</td>
</tr>
</tbody>
</table>
Definition

The zero rate threshold for online deletions, \( p_0^{(on,d)} \), is largest \( p \) where exist deterministic codes \( C \) against \( pN \) online deletions.

Theorem (Guruswami and L. '17)

\[ p_0^{(adv)} = \frac{1}{2} \text{ if and only if } p_0^{(on,d)} = \frac{1}{2}. \]

Specifically, we prove

\[ p_0^{(adv)} \leq p_0^{(on)} \leq \frac{1}{3 - 2p_0^{(adv)}}. \]

Open questions

- Can one find codes against $p$ fraction of *online* deletions when $p$ approaches $1/2$?
Outline for section 4

1. Motivating problem
2. Adversarial deletions
3. Online deletions
4. Oblivious deletions
5. Conclusion
Oblivious deletion codes need stochastic encoding

Channel knows: Enc, Dec, Message set $\mathcal{M}$, Code $C$, message $m$. Channel does not know: codeword $c$.

With deterministic code, same as adversarial deletions. Need *stochastic encoding*.

$$m = 01 \quad \text{Alice (Enc)} \quad c = 00001111 \quad \text{Channel} \quad s = 0000 \quad \text{Bob (Dec)} \quad m = ??$$
Oblivious deletion codes need stochastic encoding

Channel knows: Enc, Dec, Message set $\mathcal{M}$, Code $C$, message $m$. Channel does not know: codeword $c$.

With deterministic code, same as adversarial deletions. Need *stochastic encoding*.

$$m = 01 \rightarrow \text{Alice (Enc)} \rightarrow \text{Channel} \rightarrow \text{Bob (Dec)} \rightarrow m = ??$$

$$m = 01 \rightarrow \text{Alice (Enc)} \rightarrow \text{Channel} \rightarrow \text{Bob (Dec)} \rightarrow m = 01$$
Definition

The zero rate threshold for oblivious deletions, $p_0^{(obliv)}$, is largest $p$ s.t. exist constant rate codes $C$ against $pn$ oblivious deletions.

Q: What is $p_0^{(obliv)}$?
What is zero rate threshold for oblivious deletions?

**Definition**
The zero rate threshold for oblivious deletions, $p_{0}^{(obliv)}$, is largest $p$ s.t. exist constant rate codes $C$ against $pn$ oblivious deletions.

**Q:** What is $p_{0}^{(obliv)}$?

**Theorem (Guruswami and L. ’17 (informal))**

\[ p_{0}^{(obliv)} = 1 \]

This is an existential result.
Step 1: Average case del. code gives oblivious del. code

Oblivious deletions: “for every channel, every message decoded correctly w.h.p.” (stochastic)
Average case deletions: “for every channel, most messages decoded correctly w.p. 1.” (deterministic)

Theorem (Average case deletions, Guruswami and L. ’17)

For any $p < 1$, there exist constant rate codes $C$ for $pn$ average case deletions.
Step 1: Average case del. code gives oblivious del. code

Oblivious deletions: “for every channel, every message decoded correctly w.h.p.” (stochastic)
Average case deletions: “for every channel, most messages decoded correctly w.p. 1.” (deterministic)

Theorem (Average case deletions, Guruswami and L. ’17)
For any $p < 1$, there exist constant rate codes $C$ for $pn$ average case deletions.

Average $\implies$ Oblivious: $\text{Enc}(m)$ uniformly samples from $N^3$ codewords
Step 2: Construct average case deletion code using [BG16]

Same construction used for $p_0^{(adv)} \geq \frac{1}{3}$ proof [Bukh, Guruswami '16].

$R, K$ sufficiently large constants. For $i = 1, 2, \ldots, K$, define

$$g_i = \left(0^{R_i-1}1^{R_i-1}\right)^{L/(2^{R_i-1})}$$

$$g_1 = 010101010101010101010101010101010101\ldots$$

$$g_2 = 000111000111000111000111000111000111\ldots$$

$$g_3 = 000000000111111111100000000000001111111111\ldots$$

$$\vdots \quad \vdots$$

$$g_K = 0000000000000000000000000000000000000000\ldots$$

Codewords: concatenate $n \to \infty$ random $g_i$'s together
Decoder: Find unique codeword $c$ such that $s \subseteq c$. Give up if not unique.
(Oblivious) Deletions are poorly understood

Recall: \textit{Capacity} is optimal Rate vs. \( p \) tradeoff

Q. What is the capacity of oblivious deletions?

<table>
<thead>
<tr>
<th>Error type</th>
<th>( p_0^{(obliv)} )</th>
<th>Capacity</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substitutions</td>
<td>( 1/2 )</td>
<td>( 1 - H(p) )</td>
<td>[Lapidoth, Narayan '98]</td>
</tr>
<tr>
<td>Erasures</td>
<td>( 1 )</td>
<td>( 1 - p )</td>
<td>[Lapidoth, Narayan '98]</td>
</tr>
<tr>
<td>Deletions</td>
<td>( 1 )</td>
<td>Poor bounds</td>
<td>[This work]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error type</th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substitution</td>
<td>000</td>
<td>001</td>
</tr>
<tr>
<td>Erasure</td>
<td>000</td>
<td>00⊥</td>
</tr>
<tr>
<td>Deletion</td>
<td>000</td>
<td>00</td>
</tr>
</tbody>
</table>
Open questions

- Fact: Capacity random sub. = Cap. oblivious sub. = $1 - H(p)$.
- Capacity random deletion = Cap. oblivious deletion?
- Codes for $pn$ oblivious insertions and deletions for every $p < 1$?
- Explicit codes for oblivious deletions that are constructable, encodable, and decodable in polynomial time?
Outline for section 5

1. Motivating problem
2. Adversarial deletions
3. Online deletions
4. Oblivious deletions
5. Conclusion
Summary

We proved

- $p_{0}^{(on,d)} = \frac{1}{2}$ if and only if $p_{0}^{(adv)} = \frac{1}{2}$
- $p_{0}^{(obliv)} = 1$, i.e. $\forall p < 1$, exist codes against $pn$ oblivious deletions

Open questions

- What is the zero rate threshold, $p_{0}^{(adv)}$?
- Codes against $p$ fraction of online deletions when $p \to \frac{1}{2}$?
- Codes against $pn$ oblivious ins. and dels. for every $p < 1$?
- Efficient codes for $pn$ oblivious deletions for every $p < 1$?
- Capacity random deletion = capacity oblivious deletion?
Thanks!

\[ m = 01 \rightarrow \text{Alice (Enc)} \xrightarrow{c = 00001111} \text{Channel} \xrightarrow{s = 0000} \text{Bob (Dec)} \rightarrow m = ?? \]

\[ g_i = \left( 0^{R_{i-1}} 1^{R_{i-1}} \right)^{L/(2R_{i-1})} \]

\[ g_1 = 0101010101010101010101010101010101010101 \ldots \]

\[ g_2 = 000111000111000111000111000111000111000111 \ldots \]

\[ g_3 = 00000000011111111100000000111111111 \ldots \]

\[ \vdots \]

\[ g_K = 000000000000000000000000000000000000 \ldots \]