Deletion codes in the high-noise and high-rate regimes

Venkatesan Guruswami Carol Wang

Carnegie Mellon University

12 February 2015

The deletion model

Erasures:

? 1 0 ? 1 0 ? ? ? ? 0 1 ? ? 1 1

Symbols are lost; receiver sees "?".

Deletions:

1 0 1 0 0 1 1 1

1 0 1 0 0 1 1 1

Symbols are lost; receiver sees *nothing* (gets a subsequence of message).

The deletion model cont.

Some assumptions:

- No errors or insertions
- Receiver knows block length
- Our results will be for adversarial deletions

Why deletions?

- Natural model for asynchronous channels
- Can think of dropped packets
- Nice combinatorial questions

Deletions are easy!

1	2	3	4	5		n
m_1	m_2	m_3	m_4	m_5	$\cdots$	m_n

Reduces to erasures (easier), but alphabet is $\Omega(n)$.

Question

What can we do over constant alphabets?

Previous work

- Lots of work on constant *number* of deletions (we're interested in constant *fraction*).
- Random deletions: for deletion probability p, capacity is at least $(1-p)/9$ [Mitzenmacher and Drinea].
- Random deletions: for p going to 0, capacity is $\approx 1 - h(p)$ [Kalai, Mitzenmacher, Sudan].
- Adversarial deletions: explicit good binary codes correcting constant fraction of deletions [Schulman and Zuckerman].

What do we study?

- Goal: Understand tradeoff between redundancy (rate) and correction capability.
- For fixed deletion fraction p, what's the best rate we can get?
- Difficult even for *random* deletions.

We focus on coding for the two extremes, *low* and *high* deletion fractions.

What's possible?

Greedy construction gives:

- (High noise) There exist codes correcting a $1-\epsilon$ deletion fraction with rate $\Omega(\epsilon)$ and alphabet size $O(1/\epsilon^3)$.
- (Low noise) There exist binary codes correcting an ϵ deletion fraction with rate $\approx 1-2h(\epsilon)$.

For high noise, large alphabet is *necessary*: With $> 1/2$ deletion fraction on a binary codeword, can delete all 1's or all 0's.

Our results

Theorem (High noise)

There is an explicit code which can correct a $1 - \epsilon$ fraction of deletions with rate ϵ^2 and alphabet size $1/\epsilon^4$.

(Existential: rate ϵ, alphabet $1/\epsilon^3$.)

Theorem (High rate)

There is an explicit code which can correct a ϵ fraction of deletions with rate $\sim 1 - \sqrt{\epsilon}$ and alphabet size 2.

(Existential: $\sim 1 - \epsilon$ rate.)

Idea: concatenation

We know two kinds of deletion codes:

- Good explicit codes for large alphabets (headers)
- Good non-explicit codes for small alphabets (brute-force)

Put them together!

$$
\begin{array}{ccccccc}
B_1 & B_2 & B_3 & \cdots & B_m & & \text{(outer code)} \\
\downarrow & \downarrow & \downarrow & & \downarrow & & \\
\underbrace{\mathrm{Enc}(B_1)}_{\text{block}} & \mathrm{Enc}(B_2) & \mathrm{Enc}(B_3) & \cdots & \mathrm{Enc}(B_m) & & \text{(inner code)}
\end{array}
$$

Concatenation cont.

How to decode a concatenated code?

$$\mathrm{Enc}(B \qquad \mathrm{nc}(B_2)\mathrm{E} \qquad\qquad)\cdots\mathrm{E} \quad \mathrm{c}(B_m)$$

$$\Big\downarrow \ \text{deletions}$$

$$\mathrm{Enc}(B\mathrm{nc}(B_2)\mathrm{E})\ldots\mathrm{Ec}(B_m)$$

If we knew where blocks were, could decode.

Challenge: How to find blocks?

This talk: Different schemes for locating blocks.

High deletions

Theorem

There is an explicit code which can correct a $1 - \epsilon$ fraction of deletions with rate poly(ϵ) and alphabet size poly($1/\epsilon$).

Initial code: Concatenate a Reed-Solomon code (with headers) and a small-alphabet code against a $1 - \epsilon/2$ deletion fraction.

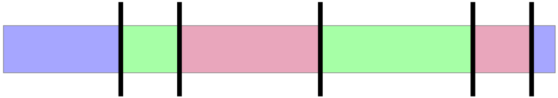
Need to modify the code to help locate blocks.

High deletions cont.

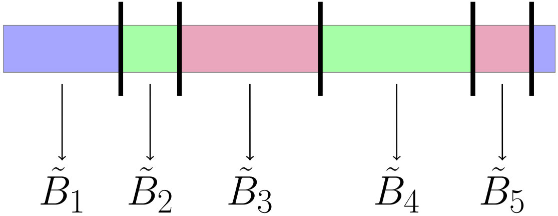
Idea: add constant-sized labels.

$$\mathrm{Enc}(B_1)\mathrm{Enc}(B_2)\mathrm{Enc}(B_3)\mathrm{Enc}(B_4)\mathrm{Enc}(B_5)\mathrm{Enc}(B_6)\cdots$$

Now receiver can use colors (labels) to guess and decode blocks.



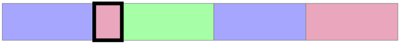
Algorithm



- Decode if “many” windows are correct
- Need to bound number of bad windows

Analysis

Idea: with $1-\epsilon$ fraction of deletions, adversary can't affect too many blocks.



Must delete $> 1-\epsilon/2$ fraction to prevent decoding.

If number of colors is poly$(1/\epsilon)$, also expensive.

High-rate binary codes

Theorem

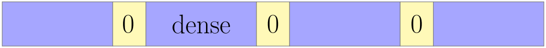
There is an explicit binary code of rate $1-\epsilon$ which can correct a poly(ϵ) fraction of deletions.

Initial code: High-rate Reed-Solomon (with headers) concatenated with inner binary code for $\sim \sqrt{\epsilon}$ deletions.

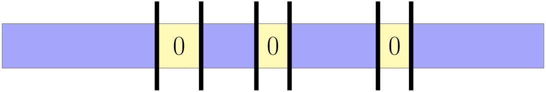
Fact: can choose inner code to be “dense”: long substrings have many 1’s.

High-rate cont.

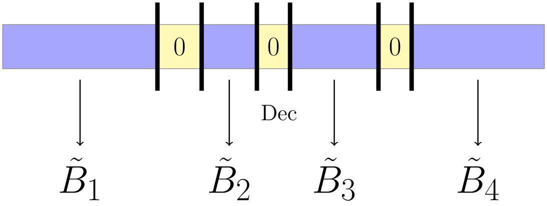
Idea: use "buffers" of 0's to separate blocks.



Dense inner code means decoder can look for long runs of 0's.



Algorithm

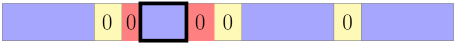


- Decode if “many” windows are correct
- Need to bound number of bad windows

Analysis

Idea: with small fraction of deletions, adversary can't affect too many blocks.

Must delete most of buffer.



Thanks to density, must delete many 1's.

Conclusion and open questions

Constructed good deletion codes for high noise and high rate, but there's still a lot we don't know.

- For binary codes, what is the highest fraction we can correct with constant rate?
- How about for fixed alphabet size k?
- Are there efficient codes of rate $1 - p - \gamma$ correcting a p fraction of deletions with alphabet only depending on γ?

Thanks! Questions?