Bounds on the size of single deletion error correcting codes

NYCS, April 14, 2023

This is a survey talk with some new results.

Seminar and friends:
Austin Anderson
Quinn Culver
Manabu Hagiwara
Ellen Hughes
Justin Kong
Kazuhisa Nakasho
J. B. Nation

Classical sources:
V. I. Levenshtein
N. J. A. Sloane
A. A. Kulkarni and N. Kiyavash

Japan

Hawaiʻi

Noisy communication channel

$\longrightarrow$

Types of error

- sent 10011
- bit-flip: received 1**1**011
- erasure: received 1**?**011
- deletion: received 1011
- insertion: received 110011

Example of SDECC

A single deletion error correcting code is capable of correcting one deletion error.

n=5

$C = \{\, 00000, 11100, 10001, 11011, 01010, 00111 \,\}$

Deletions:

- $00000 \rightarrow 0000$
- $11100 \rightarrow 1100$, 1110
- $10001 \rightarrow 0001$, 1001, 1000
- $11011 \rightarrow 1011$, 1111,1101
- $01010 \rightarrow 1010$, 0010, 0110, 0100, 0101
- $00111 \rightarrow 0111$, 0011

Fundamental results

Let $x \in 2^n$.
The **deletion surface** $S_D(x)$ is all $y \in 2^{n-1}$ that are deletions of x.
The **insertion surface** $S_I(x)$ is all $z \in 2^{n+1}$ that are insertions of x.

$C \subseteq 2^n$ is a single deletion error correcting code (SDECC) if $S_D(x) \cap S_D(x') = \varnothing$ whenever $x \neq x'$, both in C.

Lemma: $S_D(x) \cap S_D(x') = \varnothing$ iff $S_I(x) \cap S_I(x') = \varnothing$

Levenshtein: A code C is capable of correcting t deletions iff it is capable of correcting t insertions.

Levenshtein also gave a decoding algorithm to correct single deletions from $\mathrm{VT}_\ell(n)$. (Varshamov-Tenengolts codes)

Notation

A word of Hamming weight k will be denoted $x = (a_1, \ldots, a_k)$ with $a_1 < a_2 < \cdots < a_k$ giving the places where x_j is 1. There are $\binom{n}{k}$ such words.

For example, 10101000 is denoted $(1, 3, 5)$, and there are $\binom{8}{3} = 56$ words of length 8 and weight 3.

We use the function ρ where, if the representation of x is $(a_1, \ldots, a_k)$, then

$$\rho(x) = a_1 + \cdots + a_k$$

and we will consider $\rho(x) \pmod{m}$ for various m.
For example, $\rho(1, 3, 5) = 9 = 3 \pmod{6}$.

The \$64 Question

Q'n: **What is max(n), largest size of SDECC of length n?**

Conjecture: $\mathbf{max}(n) = |\,\mathrm{VT}_0(n)|$

$\mathrm{VT}_\ell(n) = \{x \in 2^n : \rho(x) = \ell \pmod{n+1}\}$

Example: $\mathrm{VT}_0(5) = \{\,(\,), (123), (15), (1245), (24), (345)\,\}$

$\mathrm{VT}_\ell(n)$ is a SDECC
(if x and x' have a common deletion, then $|\rho(x) - \rho(x')| \leq n$)

so $|\mathrm{VT}_\ell(n)|$ is a lower bound on max(n).

$$\frac{2^n}{n+1} \leq |\mathrm{VT}_0(n)| \leq \ \max(n) \leq \frac{2^n}{n}$$

Fun facts about VT codes

$$|\mathrm{VT}_\ell(n)| \approx \frac{2^n}{n+1}$$

$$|\mathrm{VT}_0(n)| \geq |\mathrm{VT}_\ell(n)| \geq |\mathrm{VT}_1(n)|$$

$|\mathrm{VT}_0(n)| = |\mathrm{VT}_1(n)|$ iff n+1 is a power of 2

$$|\mathrm{VT}_0(n)| = \frac{1}{2(n+1)} \sum_{d|n=1, d \ odd} \phi(d) 2^{\frac{n+1}{d}}$$

Each $\mathrm{VT}_\ell(n)$ is a perfect code

max(n) = $|\mathrm{VT}_0(n)|$ for $n \leq 10$
(Sloane, Applegate, Butenko et al.)

Exciting new result of No, Nakasho

n	$\lvert VT_0(n)\rvert$	$\max(n)$	UB
2	2	2	2
3	2	2	2
4	4	4	4
5	6	6	6
6	10	10	10
7	16	16	16
8	30	30	30
9	52	52	52
10	94	94	94
11	172	172	**172**
12	316	?	320
13	586	?	593
14	1096	?	1104
15	2048	?	2184

Non-uniqueness

Note that $\mathrm{VT}_0(n)$ is not unique as the largest known SDECC of length n.

$$x = 000101111\cdots$$
$$y = 000001111\ldots$$

If x is in a SDECC, then it can be replaced by y to obtain another SDECC of the same size since $S_D(y) \subset S_D(x)$

(This observation of Sloane has been generalized by Kondo)

Make a graph G = (V,E)

- V=2^n, all binary words of length n
- (u, v) is an edge if u and v have a common deletion

E.g., with n=3, the vertices 000, 100, 010, 001 would be pairwise connected by edges because they have the common deletion 00.

SDECCs correspond to independent sets in G

Given a SDECC C, let

$$x_j = \begin{cases} 1 & \text{if } j \in C, \\ 0 & \text{otherwise} \end{cases}$$

Size problem: Maximize $\sum_{i \in 2^n} x_i$ subject to

$$(\dagger) \quad \forall i \in 2^n \;\; x_i \in \{0, 1\} \text{ and } x_i + x_j \leq 1 \text{ whenever } (i, j) \in E$$

How it got down to 172

Size problem: Maximize $\sum_{i \in 2^n} x_i$ subject to

$$(\dagger) \quad \forall i \in 2^n \;\; x_i \in \{0,1\} \text{ and } x_i + x_j \leq 1 \text{ whenever } (i,j) \in E$$

Change it to a linear programming problem:
Maximize $\sum_{i \in 2^n} x_i$ subject to

$$(\ddagger) \quad \forall i \in 2^n \;\; 0 \leq x_i \leq 1 \text{ and } x_i + x_j \leq 1 \text{ whenever } (i,j) \in E$$

Actually, that's not good enough. You have to use Mixed Integer Programming with ($\dagger$) for some edges and ($\ddagger$) for others. Using the graph for $n = 11$:

Albert No (2019): $\sum x_i \leq 173.99$

Kazuhisa Nakasho (2023): $\sum x_i \leq 172.99$

Hence max(11)=172

k-of-n SDECCs

C is a **k-of-n** code if every $x \in C$ has Hamming weight k

Example: a 3-of-8 SDECC with 10 codewords

$$
\begin{aligned}
C =& (123), (345), (246), (156), (237), (147), (567), (138), (468), (378) \\
=& 11100000, 00111000, 01010100, 10001100, 01100010, \\
& 10010010, 00001110, 10100001, 00010101, 00100011
\end{aligned}
$$

$\max_k(n)$ is the maximum size of a k-of-n SDECC

2-of-n SDECCs

$$\max_2(n) = \left\lfloor \frac{3n-2}{4} \right\rfloor$$

n	$\max_2(n)$
4	2
5	3
6	4
7	4
8	5
9	6
10	7
11	7
12	8
13	9

$$(1,2) \quad (3,4) \quad (2,5) \quad (5,6) \quad (7,8) \quad (6,9) \quad (9,10) \quad \ldots$$

$$C = \{(i,j) \in 2^n : i + j = 3 \pmod 4 \text{ and } j - i \leq 3\}.$$

Sketch of proof of upper bound for $\max_2(n)$

Let C be a 2-of-n SDECC of length n.

- a *good* codeword is of the form $(k, k+1)$
- a *bad* codeword is of the form (k, b) with $b > k+1$

so that $|C| = g + b$.
No two codewords have a common deletion, and you cannot have consecutive good codewords.

$$g \leq \frac{n}{2}$$

$$g + 2b \leq n - 1$$

(RHS is the number of weight 1 words of length n-1)

Adding, we get

$$2g + 2b \leq \frac{3n-2}{2}$$

whence

$$|C| \leq \frac{3n-2}{4}$$

VT-like k-of-n SDECCs

Following R. Graham and Sloane:

$J(k, \ell, m, n) = \{x \in 2^n : \mathrm{wt}(x) = k \text{ and } \rho(x) = \ell \pmod{m}\}$.

For $2 \leq k \leq n/2$,
$J(k, \ell, n-k+1, n)$ is a k-of-n SDECC

Example: $n = 8$, $k = 3$, $n - k + 1 = 6$, $\ell = 0$ gives the 10-element code

$(123), (345), (246), (156), (237), (147), (567), (138), (468), (378)$

Lower bound: $\dfrac{\binom{n}{k}}{n-k+1} \leq |J(k, \ell, n-k+1, n)| \leq \max_k(n)$

Question: When is $|\,\mathrm{J}(k, \ell, n-k+1, n)|$ constant for different ℓ?

Answer: If $k = p^s$ is a prime power, then exactly when $n \neq -1 \pmod{p}$. For composites, when that holds for all prime power factors of k.

Upper bound for $\max_k(n)$

For $2 \leq k \leq n/2$,

$$\frac{\binom{n}{k}}{n-k+1} \leq$$

$$\max_k(n) \leq \frac{1}{k}\binom{n-k+1}{k-1} + \frac{k-1}{k}\binom{n-k+1}{k-2} + O(n^{k-3})$$

3-of-n SDECCs

$$\frac{n^2 - n}{6} \leq \max_3(n) \leq \frac{n^2 - 3}{6}$$

n	LB	search	$\max_3(n)$	UB
6	5	5	5	5
7	7	7	7	7
8	10	10	10	10
9	12	13	13	13
10	15	16	16	16
11	19	19	19	19
12	22	**23**	23	23
13	26	**27**	27	27
14	31		?	32
15	35		?	37

The computer search values at n=12, 13 are larger than any $|J(3, \ell, n-2, n)|$

4-of-n SDECCs

$$\frac{n(n-1)(n-2)}{24} \leq \max_4(n) \leq \frac{4n^3 - 9n^2 + 2n - 48}{96}$$

n	LB	search	$\max_4(n)$	UB
8	14	14	14	15
9	22	–	22	22
10	30	–	**?**	31
11	43		43	43
12	55		?	58
13	73		?	75
14	91		?	95
15	116		?	118

5-of-n SDECCs

$$\frac{n(n-1)(n-2)(n-3)}{120} \leq \max_5(n) \leq \frac{n^4 - 6n^3 + 16n^2 - 34n - 25}{120}$$

n	LB	UB
10	42	43
11	66	68
12	99	101
13	143	146
14	201	204
15	273	278

Remember

The main question remains:

Is max(n) = $|VT_0(n)|$?

Alternating SDECCs

If C_k is a k-of-n SDECC, then take $C = C_0 \cup C_2 \cup C_4 \cup \cdots$

n	alternating	$\lvert VT_0(n) \rvert$	ratio
6	10	10	1.00
7	13	16	.78
8	26	30	.87
9	43	52	.83
10	72	94	.77
11	137	172	.79
12	260	316	.82
13	469	586	.80
14	865	1096	.79
15	1647	2048	.80
20	$41,940$	$49,934$	.84
30	$29,633,046$	$34,636,832$	.86
40	2.3615×10^{10}	2.6817×10^{10}	.88
80	1.3630×10^{22}	1.49250×10^{22}	.91

Try #2

The global approach of the last slide was naive and optimistic. One can play this game locally and still lose.

Example: $VT_4(12)$ has 22 codewords of weight 6 with $\rho(x) = 30$. We can add two words of weight 6 with $\rho(x) = 23$, viz., (123458) and (123467). But then we must remove from $VT_4(12)$ two words of weight 7 with $\rho(x) = 30$, (1234578) and (1234569), and two words of weight 5 with $\rho(x) = 13$, (12347) and (12356).

In this way we obtain a SDECC of size $315 + 2 - 4 = 313$.

Try #3

Let Top be all $x \in \mathrm{VT}_0(n)$ with wt(x) ≥ 4 and do a computer search for a set Bottom of words of weight ≤ 3 to find codes $C = \mathrm{Top} \cup \mathrm{Bottom}$ such that C is a SDECC with $|C| \geq |\mathrm{VT}_0(n)|$.

For n=12, 13 and 14, this search yields only $\mathrm{VT}_0(n)$.

The program is slow but we are still looking.

k-of-n DDECCs

We can ask the same questions about codes that correct multiple deletion/insertion errors.

A double deletion error correcting code (DDECC) is capable of correcting two deletion/insertion errors.

Example: 3-of-12 DDECC

$$(1, 2, 3)\ (3, 4, 6)\ (2, 6, 7)\ (7, 8, 9)\ (6, 9, 11)\ (10, 11, 12)$$

Let $\max_k^2(n)$ be the maximum size of a DDECC of length n and Hamming weight k.

3-of-n DDECCs

$$\frac{5n-6}{9} \leq \max_3^2(n) \leq \frac{7n-15}{9}$$

n	max
5	1
6	2
7	3
8	3
9	4
10	4
11	5
12	6
13	6

$(1,2,3)$ $(3,4,6)$ $(2,6,7)$ $(7,8,9)$ $(6,9,11)$ $(10,11,12)$

Quantum deletion codes

Quantum communication channels use *qubits* (think photons) instead of bits for messages.

Manabu Hagiwara has found a way to construct quantum deletion codes based on certain classical codes, such as Reed-Solomon codes. These codes can

- achieve any code rate < 1, and
- correct multiple quantum deletion errors (think lost photons).

Mahalo!