Deletion codes (Will C, Xiaoming)

Will C, Xiaoming

October 13, 2024

1 Deletion Code Problems

- It is known $|C| \approx \frac{2^n}{n+1}$. Give an intuitive explanation why this makes sense.
  ANS: All possible combinations of C is equal to 2^n because each of the n elements in C can be a 0 or 1. However, in the definition of C, we see that $x_1 + 2x_2 + ... + nx_n \equiv 0 \pmod{n+1}$. For this constraint to hold, $|C|$ is reduced by a factor of n+1, because now only combos of 0s and 1s that satisfy our constraint are allowed. $|C|$ is only $\approx \frac{2^n}{n+1}$ due to cases when $\frac{2^n}{n+1}$ is not an integer, e.g. $n = 4$.

- Q: Why does the factor of $n+1$ make sense given our constraint?

- Show that $|C| = \frac{2^n}{n+1}$ when n is one less than a power of 2.
  ANS: Start by letting $n = 2^t - 1$. Next we rearrange the terms in the syndrome of C, $x_1 + 2x_2 + ... + nx_n \equiv 0 \pmod{n+1}$, such that all the constants that are a power of 2 are written first. $x_1 + 2x_2 + 4x_4 + 8x_8 + ... + 2^{t-1}x_{2^{t-1}} + 3x_3 + 5x_5 + 6x_6 + ... \equiv 0 \pmod{2^t}$. Looking at the non-power of 2 terms, we see there's 2^{n-t} choices for each x_i term here, and there is one unique way of choosing each of the x_i terms paired with powers of two in order to make the entire summation with any choice of the non-power of two x_i terms congruent to 0. So $|C|$ is $\frac{2^n}{2^t} = \frac{2^n}{n+1}$ when $n = 2^t - 1$

- Q: Is the grammar, "syndrome of C"?

- Show that C *corrects one deletion.*
  ANS: For all $z_1, z_2, ..., z_{n-1} \in \{0,1\}^{n-1}$, one of the following is in C:
  $0, z_1, z_2, ..., z_{n-1}$
  $1, z_1, z_2, ..., z_{n-1}$
  $z_1, 0, z_2, ..., z_{n-1}$
  $z_1, 1, z_2, ..., z_{n-1}$
  ...
  $z_1, z_2, ..., z_{n-1}, 0$
  $z_1, z_2, ..., z_{n-1}, 1$
  If we compare any two of these codes, we can deduce that they either have different syndromes of C or $z_1 = z_2 = ... = z_{n-1} = 0$. If we plug these codes into the syndrome of C we get $0 + 2z_1 + 3z_2 + ... = nz_{n-1} \equiv 0 \pmod{n+1}$ and $z_1 + 2z_2 + ... + (n-1)z_{n-1} \equiv 0 \pmod{n+1}$. Subtracting these two we get $z_1 + z_2 + ...z_{n-1} \equiv 0 \pmod{n+1}$. This implies $z_1 = z_2 = ... = z_{n-1} = 0$.

- Show that, for n sufficiently large, any set C that corrects one deletion must have size at most $|C| \leq \mathcal{O}\left(\frac{\alpha \cdot 2^n}{n}\right)$. That is, show that $|C| \leq \frac{\alpha \cdot 2^n}{n}$ for some constant α independent of n. (For example, $\alpha = 10^{10}$).

  Proof by contradiction. Take $\alpha = 1000$, FSOC assume there exist a code C that correct 1 deletion, and $|C| > 1000 \cdot \frac{2^n}{n}$. Let S be the set of strings with at least n/4 runs, then we have $|C| = |C \cap S| + |C \cap S^C|$. Note that

$$|S^C| = 2 \cdot \sum_{i=0}^{\frac{n}{4}-1} \binom{n-1}{i} \leq 2 \cdot \sum_{i=0}^{\frac{n}{4}} \binom{n}{i} \leq 2^{H(1/4)\cdot n}$$

$$\implies |C \cap S^C| \leq 2^{H(1/4)\cdot n}$$

Taking $C \cap S$, since every string in the intersection has at least n/4 runs, then each string has at least n/4 possible substring of length n-1. So

$$\#\text{ of possible substrings} \geq |C| \cdot \frac{n}{4} > 1000 \cdot \frac{2^n}{n} \cdot \frac{n}{4} = 250 \cdot 2^n > 2^n > 2^{n-1}$$

which means that there has two be at least two string in $C \cap S$ with the same substring, which contradicts the assumption that C corrects 1 deletion, so we have

$$|C \cap S| \leq 1000 \cdot \frac{2^n}{n}$$

Thus,

$$|C| = |C \cap S| + |C \cap S^C| \leq 1000 \cdot \frac{2^n}{n} + 2^{H(1/4)\cdot n} \leq \mathcal{O}(\frac{2^n}{n})$$

□

- The *redundancy* of a code $C \subseteq \{0,1\}^n$ is equal to $n - \log_2 |C|$. Show that the following hold as a corollary of the above arguments:
  - For any integer n, there exists a code C correcting 1 deletion with redundancy at most $\log_2 n + \mathcal{O}(1)$.
    [not sure if this is correct]

    Proof. Since $|C| \approx \frac{2^n}{n+1} = \frac{2^n}{n} \cdot \frac{n}{n+1}$, then

    $$\log_2(|C|) \approx n - \log_2(n) - \mathcal{O}(1)$$
    $$\implies \text{Redundancy} = n - \log_2(|C|) \approx \log_2(n) + \mathcal{O}(1)$$

    □

  - For any integer n, a code correcting 1 deletion needs redundancy at least $\log_2 n - \mathcal{O}(1)$.

    Proof. Fix $n \in \mathbb{N}$, and let C corrects 1 deletion. Since $|C| \leq \mathcal{O}(\frac{2^n}{n})$, say $|C| \leq \frac{\alpha \cdot 2^n}{n}$ with $\alpha = 1000$, then

    $$\log_2(|C|) \leq \log_2(\frac{\alpha \cdot 2^n}{n}) = \log_2(\alpha) + n - \log_2(n) = \mathcal{O}(1) + n - \log_2(n)$$
    $$\text{Redundancy} = n - log_2(|C|) \geq n - \mathcal{O}(1) - n + \log_2(n) = \log_2(n) - \mathcal{O}(1)$$

    □

- Show that, for any fixed positive integer $t = 1, 2, 3, \ldots$, there exists a code C correcting t deletions with redundancy $2t \log_2 n + \mathcal{O}_t(1)$. (The $\mathcal{O}_t(1)$ can suppress constants that depend on t, like t^2)

  Proof by construction. Fix $t \in \mathbb{N}$. Start with an empty set C, and we want to add strings into C such that all strings in C are distance $2t$ away. For each string in C, the number of strings we can get from $2t$ insdels is $\binom{n}{t} \cdot (n - t + 1)^t \cdot 2^t$. We want to add strings until

  $$|C| \cdot \binom{n}{t} \cdot (n - t + 1)^t \cdot 2^t > 2^n$$

  This implies:

  $$|C| > \frac{2^n}{\binom{n}{t} \cdot (n - t + 1)^t \cdot 2^t} > \frac{2^n}{n^t \cdot n^t \cdot 2^t} = \frac{2^n}{n^{2t} \cdot 2^t}$$

  Taking the log of both side:

  $$\log_2(|C|) > \log_2\left(\frac{2^n}{n^{2t} \cdot 2^t}\right) = n - 2t \log_2(n) - t$$

Hence, the redundancy is:

$$\text{Redundancy} = n - \log_2(|C|) < n - (n - 2t\log_2(n) - t) = 2t\log_2(n) + t = 2t\log_2(n) + \mathcal{O}_t(1)$$

Thus showing we can have a code C correcting t deletions with redundancy of $2t\log_2(n) + \mathcal{O}_t(1)$. □

- Show that, for any fixed positive integer $t = 1, 2, 3, \ldots$, every code C correcting t deletions must have redundancy at least $t\log_2 n - O_t(1)$.

Proof. Fix t $\in N$, and let C be the set of length n strings that correct t deletions. Since for each string in C, there are $\binom{n}{t}$ possible length $(n-t)$ substring, then

$$\binom{n}{t} \cdot |C| \leq 2^{n-t} \Longrightarrow |C| \leq \frac{2^{n-t}}{\binom{n}{t}} \leq \frac{2^{n-t} \cdot t!}{n^t} = \mathcal{O}_t(\frac{2^n}{n^t}).$$

Taking log of both side, then

$$\log_2(|C|) \leq n - t\log_2(n) + \mathcal{O}_t(1).$$

meaning that

$$\textit{Redundancy} = n - log_2(|C|) \geq n - n + t\log_2(n) - \mathcal{O}_t(1) = t\log_2(n) - \mathcal{O}_t(1)$$

□

- Do a literature search for the best redundancy for codes correcting deletions. The following papers will be good references:

t	Lower bound	Upper Bound (Existential)	Upper Bound (Constructive)
1	$\log_2 n$	$\log_2 n$	$\log_2 n$
2	$2\log_2 n$	$4\log_2 n$	$4\log_2 n + \mathcal{O}(\log\log n)$
$t \geq 3$	$t\log_2 n$	$2t\log_2 n$	$4t\log_2 n + o(\log n)$

2 Homework Problem 1

1. Given that

$$e^t = \sum_{i=0}^{\infty} \frac{t^i}{i!} = 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \ldots$$

Then

$$e^x - (1+x) = \sum_{i=0}^{\infty} \frac{x^i}{i!} - (1+x) = \sum_{i=2}^{\infty} \frac{x^i}{i!}$$

Since $2^i - 1 \leq i!$ and $x^i \leq 1$ for all $-1 \leq x \leq 1$,

$$\sum_{i=2}^{\infty} \frac{x^i}{i!} \leq \sum_{i=2}^{\infty} \frac{x^i}{2^{i-1}} = x^2 \sum_{i=1}^{\infty} \frac{x^{i-1}}{2^i} \leq x^2 \sum_{i=1}^{\infty} \frac{1}{2^i} = x^2$$

So that

$$e^x \leq 1 + x + x^2 \quad \text{for all} \quad -1 \leq x \leq 1.$$

Thus as $x \to 0$,

$$\Longrightarrow e^x \leq 1 + x + x^2 = 1 + x + \mathcal{O}(x^2)$$

2. (a) Fix $k \in \mathbb{N}^+$, then

$$e^k = \sum_{i=0}^{\infty} \frac{k^i}{i!} = 1 + k + \frac{k^2}{2!} + \cdots + \frac{k^k}{k!} + \cdots < \frac{k^k}{k!}$$

(b) Fix $0 < k < n$.

Lower Bound: WTS $\left(\frac{n}{k}\right)^k \leq \binom{n}{k}$.

$$\begin{aligned}\binom{n}{k} &= \frac{n!}{k!(n-k)!} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k(k-1)(k-2)\cdots 1} \\ &= \frac{n}{k} \cdot \frac{n-1}{k-1} \cdots \frac{n-k+1}{1} \\ &\leq \frac{n}{k} \cdot \frac{n}{k} \cdots \frac{n}{k} = \left(\frac{n}{k}\right)^k\end{aligned}$$

Upper Bound: WTS $\binom{n}{k} < \left(\frac{en}{k}\right)^k$.
By Stirling's Formula, we get that

$$e \cdot \left(\frac{k}{e}\right)^k \leq k! \implies \frac{1}{k!} \leq \frac{1}{e} \cdot \left(\frac{e}{k}\right)^k < \left(\frac{e}{k}\right)^k$$

Then

$$\binom{n}{k} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k!} \leq \frac{n^k}{k!} < n^k \cdot \left(\frac{e}{k}\right)^k = \frac{n^k \cdot e^k}{k^k} = \left(\frac{en}{k}\right)^k$$

Thus,

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} < \left(\frac{en}{k}\right)^k$$

3 (a)

$$\sqrt{n+1} - \sqrt{n} = \Theta\left(\frac{1}{\sqrt{n}}\right)$$

Take $C_1 = 0.25$, and $C_2 = 2$, Then we can get the upper bound as $\sqrt{n+1} - \sqrt{n} < \frac{2}{\sqrt{n}}$, since $\sqrt{n} \cdot \left(\sqrt{n+1} - \sqrt{n}\right) \leq \sqrt{n+1} \cdot \sqrt{n+1} - \sqrt{n} \cdot \sqrt{n} = n + 1 - n = 1 < 2$. And similarly for large n (e.g.$n > 100$), $\left(\sqrt{n+1} - \sqrt{n}\right) \cdot \left(\sqrt{n+1} + \sqrt{n}\right) = 1 > \frac{\left(\sqrt{n+1} + \sqrt{n}\right)}{4\sqrt{n}}$, $\implies \sqrt{n+1} - \sqrt{n} > \frac{1}{4\sqrt{n}}$. Thus $\frac{1}{4\sqrt{n}} \leq \left(\sqrt{n+1} - \sqrt{n}\right) \leq \frac{2}{\sqrt{n}}$

(b)

$$\sum_{i=1}^{n} i^3 = \Theta\left(n^4\right)$$

Noting that $\int_0^n x^3 dx < \sum_{i=1}^n i^3 < \int_0^{n+1} x^3 dx$. Then for $n > 2$, $\frac{n^4}{4} < \sum_{i=1}^n i^3 < \frac{(n+1)^4}{4} < n^4$.

(c)

$$\frac{1}{1+\varepsilon} - (1-\varepsilon) = \Theta(\varepsilon^2)$$

Consider $(1+\varepsilon) \cdot \left(\frac{1}{1+\varepsilon} - (1-\varepsilon)\right) = 1 - (1-\varepsilon^2) = \varepsilon^2 \implies \frac{1}{1+\varepsilon} - (1-\varepsilon) = \frac{\varepsilon^2}{1+\varepsilon}$. Then for $0 \leq \varepsilon \leq 1$, $\frac{1}{2} \leq \frac{1}{1+\varepsilon} \leq 1 \implies \frac{1}{2}\varepsilon^2 \leq \frac{1}{1+\varepsilon} - (1-\varepsilon) = \frac{\varepsilon^2}{1+\varepsilon} \leq \varepsilon^2$.

(d)

$$\sum_{i=n+1}^{n^2} \frac{1}{i} = \Theta(ln(n))$$

Consider $\int_{n+1}^{n^2+1} \frac{1}{x} dx < \sum_{i=n+1}^{n^2} \frac{1}{i} < \int_n^{n^2} \frac{1}{x} dx$, then for $n \geq 1$, $\frac{1}{2} ln(n) < ln(\frac{n^2+1}{n+1}) < \sum_{i=n+1}^{n^2} \frac{1}{i} < ln(\frac{n^2}{n}) = ln(n)$

(e)

$$log_2\left(\frac{1}{\frac{1}{2}-\varepsilon}\right) = 1 + \frac{2\varepsilon}{ln(2)} + \Theta(\varepsilon^2)$$

Assume $\varepsilon \to 0^+$. Simplify the equation to get $log_2\left(\frac{1}{\frac{1}{2}-\varepsilon}\right) = log_2\left(\frac{2}{1-2\varepsilon}\right) = log_2(2) - log_2(1-2\varepsilon) = 1 - log_2(1-2\varepsilon) = 1 - \frac{1}{ln(2)}ln(1-2\varepsilon)$. Then apply the Taylor expansion of $ln(x)$ centered around $x = 1$ with $x = 1-2\varepsilon$, $\Longrightarrow$

$$1 - \frac{1}{ln(2)}ln(1-2\varepsilon) = 1 - \frac{1}{ln(2)}\left(((1-2\varepsilon)-1) - \frac{1}{2!}((1-2\varepsilon)-1)^2 + \frac{1}{3!}((1-2\varepsilon)-1)^3 - \dots\right)$$

$$= 1 + \frac{1}{ln(2)}\left((2\varepsilon) + \frac{(2\varepsilon)^2}{2!} + \frac{(2\varepsilon)^3}{3!} + \dots\right) = 1 + \frac{2\varepsilon}{ln(2)} + \Theta(\varepsilon^2)$$

3. (a) I used Desmos to plot this.

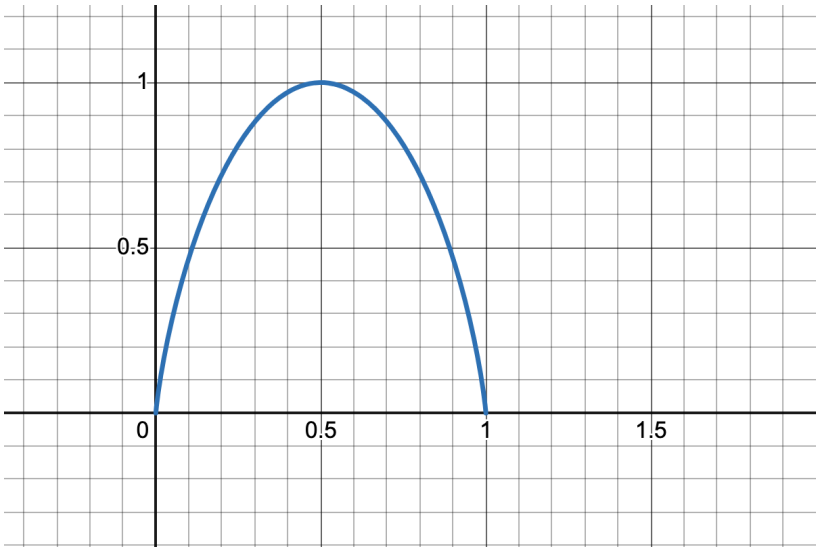

Figure 1: This is the plot for the binary entropy function for p between 0 and 1.

(b) WTS:

$$p\log_2\left(\frac{1}{p}\right) \le H_2(p) \le p\log_2\left(\frac{e}{p}\right) \text{ for all } 0 \le p \le \frac{1}{2}$$

Fix $0 \le p \le \frac{1}{2}$.
Lower Bound: Since $0 \le p \le \frac{1}{2}$,

$$\frac{1}{2} \le (1-p) \le 1 \implies (1-p)\log_2\left(\frac{1}{1-p}\right) \ge 0$$

then

$$p\log_2\left(\frac{1}{p}\right) \le p\log_2\left(\frac{1}{p}\right) + (1-p)\log_2\left(\frac{1}{1-p}\right) = H_2(p)$$

Upper Bound:
WTS:

$$H_2(p) = p\log_2(p) + (1-p)\log_2\left(\frac{1}{1-p}\right) \le p\log_2\left(\frac{e}{p}\right) = p\log_2(e) + p\log_2\left(\frac{1}{p}\right)$$

This is equivalent to:

$$(1-p)\log_2\left(\frac{1}{1-p}\right) \leq p\log_2(e)$$

$$(1-p)\log_2(1-p) \leq p\log_2(e)$$

$$\left(\frac{p-1}{p}\right)\log_2(1-p) \leq \log_2(e)$$

$$\log_2(1-p) - \frac{1}{p}\log_2(1-p) \leq \log_2(e)$$

$$-\frac{1}{p}\log_2(1-p) \leq \log_2(e) - \log_2(1-p)$$

$$-\frac{1}{p}\log_2(1-p) \leq \log_2\left(\frac{e}{1-p}\right)$$

$$-\log_2(1-p) \leq p\log_2\left(\frac{e}{1-p}\right)$$

$$\frac{1}{1-p} \leq \left(\frac{e}{1-p}\right)^p$$

Since $0 \leq p \leq \frac{1}{2}$, $(1-p) > (1-p)^p \implies \frac{1}{1-p} \leq (\frac{1}{1-p})^p$.
And $1 \leq e^p$, so $\frac{1}{1-p} \leq (\frac{e}{1-p})^p \implies H_2(p) \leq p\log_2(\frac{e}{p})$
Thus,

$$p\log_2\left(\frac{1}{p}\right) \leq H_2(p) \leq p\log_2\left(\frac{e}{p}\right) \text{ for all } 0 \leq p \leq \frac{1}{2}$$

Then, as $p \to 0^+$,

$$1 = \frac{p\log_2\left(\frac{1}{p}\right)}{p\log_2\left(\frac{1}{p}\right)} \leq \frac{H_2(p)}{p\log_2\left(\frac{1}{p}\right)} \leq \frac{p\log_2\left(\frac{e}{p}\right)}{p\log_2\left(\frac{1}{p}\right)}.$$

Since the lower bound is 1, consider the upper bound:

$$\frac{p\log_2\left(\frac{e}{p}\right)}{p\log_2\left(\frac{1}{p}\right)} = \frac{\log_2\left(\frac{e}{p}\right)}{\log_2\left(\frac{1}{p}\right)} = \frac{\log_2(e) + \log_2\left(\frac{1}{p}\right)}{\log_2\left(\frac{1}{p}\right)} = 1 + \frac{\log_2(e)}{\log_2\left(\frac{1}{p}\right)}$$

Since as $p \to 0^+$, $\log_2\left(\frac{1}{p}\right) \to \infty$, we have:

$$\frac{p\log_2\left(\frac{e}{p}\right)}{p\log_2\left(\frac{1}{p}\right)} = 1 + \frac{\log_2(e)}{\log_2\left(\frac{1}{p}\right)} \to 1.$$

Thus,

$$\frac{H_2(p)}{p\log_2\left(\frac{1}{p}\right)} \to 1 \implies H_2(p) \sim p\log_2\left(\frac{1}{p}\right) \text{ as } p \to 0^+.$$

(c) Since

$$\log_2\left(\frac{1}{\frac{1}{2}-\varepsilon}\right) = 1 + \frac{2\varepsilon}{\ln(2)} + \Theta(\varepsilon^2),$$

and

$$\log_2\left(\frac{1}{\frac{1}{2}+\varepsilon}\right) = 1 - \frac{2\varepsilon}{\ln(2)} + \Theta(\varepsilon^2),$$

then as $\varepsilon \to 0^+$:

$$\begin{aligned} H_2\left(\frac{1}{2}-\varepsilon\right) &= \left(\frac{1}{2}-\varepsilon\right)\log_2\left(\frac{1}{\frac{1}{2}-\varepsilon}\right) + \left(\frac{1}{2}+\varepsilon\right)\log_2\left(\frac{1}{\frac{1}{2}+\varepsilon}\right). \\ &= \left(\frac{1}{2}-\varepsilon\right)\left(1 + \frac{2\varepsilon}{\ln(2)} + \Theta(\varepsilon^2)\right) + \left(\frac{1}{2}+\varepsilon\right)\left(1 - \frac{2\varepsilon}{\ln(2)} + \Theta(\varepsilon^2)\right). \\ = \left(\frac{1}{2}-\varepsilon\right) &+ \left(\frac{2\varepsilon}{\ln(2)} - \frac{4\varepsilon^2}{\ln(2)} + \Theta(\varepsilon^2)\right) + \left(\frac{1}{2}+\varepsilon\right) - \left(\frac{2\varepsilon}{\ln(2)} + \frac{4\varepsilon^2}{\ln(2)} + \Theta(\varepsilon^2)\right). \\ &= 1 - \frac{4\varepsilon^2}{\ln(2)} + \Theta(\varepsilon^2). \end{aligned}$$

Thus,

$$H_2\left(\frac{1}{2}-\varepsilon\right) = 1 - \Theta(\varepsilon^2).$$

4. (a) Fix $1 \le i \le k \le \frac{n}{2}$, then

$$\frac{\binom{n}{i}}{\binom{n}{i-1}} = \frac{\frac{n!}{(i)!(n-i)!}}{\frac{n!}{(i-1)!(n-i+1)!}} = \frac{(i-1)!(n-i+1)!}{i!(n-i)!} = \frac{n-i+1}{i}$$

Since $i \le k \le \frac{n}{2}$, $\implies n-i+1 \ge \frac{n}{2}+1 \ge k \ge i$, then

$$\frac{\binom{n}{i}}{\binom{n}{i-1}} = \frac{n-i+1}{i} \ge 1$$

Thus,

$$\binom{n}{0} \le \binom{n}{1} \le \binom{n}{2} \le \cdots \le \binom{n}{k}.$$

(b) WTS: $\theta^{pn} \cdot V(n,k) \le (1+\theta)^n$
Note that $p = k/n$, then fix$k \le n/2$, and

$$\theta^{pn} \cdot V(n,k) = \theta^k \cdot V(n,k) = \theta^k \cdot \sum_{i=0}^{k}\binom{n}{i}$$

As $0 < \theta \le 1$,

$$\begin{aligned} \implies \theta^k \cdot \sum_{i=0}^{k}\binom{n}{i} &= \theta^k\binom{n}{0} + \theta^k\binom{n}{1} + \theta^k\binom{n}{2} + \cdots + \theta^k\binom{n}{k} \\ &\le 1^n\binom{n}{0} + \theta^1 1^{n-1}\binom{n}{1} + \theta^2 1^{n-2}\binom{n}{2} + \cdots + \theta^k 1^{n-k}\binom{n}{k} \\ &\le 1^n\binom{n}{0} + \theta^1 1^{n-1}\binom{n}{1} + \theta^2 1^{n-2}\binom{n}{2} + \cdots + \theta^k 1^{n-k}\binom{n}{k} + \cdots + \theta^n\binom{n}{n} \\ &= (1+\theta)^n \end{aligned}$$

3 Homework Problem 2

1. (a) According to the Chernoff bound, we have the following:

$$\mathbf{Pr}[\bar{X} \geq (1+\varepsilon)p] = \mathbf{Pr}[X \geq (1+\varepsilon)\mu] \leq \exp\left(-\frac{\varepsilon^2}{2+\varepsilon}\mu\right)$$

and

$$\mathbf{Pr}[\bar{X} \leq (1-\varepsilon)p] = \mathbf{Pr}[X \leq (1-\varepsilon)\mu] \leq \exp\left(-\frac{\varepsilon^2}{2}\mu\right)$$

Then,

$$\mathbf{Pr}[|\bar{X}-p| > \varepsilon] \leq \mathbf{Pr}[|\bar{X}-p| \geq \varepsilon p] = \mathbf{Pr}[|X-\mu| \geq \varepsilon\mu] \leq 2 \cdot \exp\left(-\frac{\varepsilon^2}{2+\varepsilon}\mu\right) \leq 2 \cdot \exp\left(-\frac{\varepsilon^2}{3}\mu\right)$$

If $n = O\left(\frac{1}{\varepsilon^2}\log(1/\delta)\right)$, say for $C = 100$, $n = C \cdot \frac{1}{\varepsilon^2}\log(1/\delta)$, then

$$\begin{aligned}
\mathbf{Pr}[|\bar{X}-p| > \varepsilon] &\leq 2 \cdot \exp\left(-\frac{\varepsilon^2}{3}\mu\right) \\
&= 2 \cdot \exp\left(-\frac{\varepsilon^2}{3}np\right) \\
&= 2 \cdot \exp\left(-\frac{\varepsilon^2}{3}\left(C \cdot \frac{1}{\varepsilon^2}\log(1/\delta)\right)p\right) \\
&= 2 \cdot \left(\frac{1}{\delta}\right)^{-\frac{C}{3}} \\
&= 2 \cdot \delta^{\frac{C}{3}} \\
&\leq \delta
\end{aligned}$$

(b) Let $p = \mathbf{Pr}[Y \leq m]$, and we want to show $\frac{1}{2} - \varepsilon \leq \mathbf{Pr}[Y \leq m] = p \leq \frac{1}{2} + \varepsilon$. For $i \in [n]$, define $Z_i = 1$ if $Y_i \leq m$, and $Z_i = 0$ otherwise, then Z_i are i.i.d. Bernoulli random variables with mean p.
Assume $n = O\left(\frac{1}{\varepsilon^2}\log(1/\delta)\right)$. Let $\bar{Z} = \frac{1}{n}\sum_{i=1}^{n} Z_i$, then from (a),

$$\mathbf{Pr}[|\bar{Z}-p| > \varepsilon] \leq \delta \Longrightarrow \mathbf{Pr}[p-\varepsilon \leq \bar{Z} \leq p+\varepsilon] \geq 1-\delta \Longrightarrow \mathbf{Pr}\left[\left|\bar{Z}-\frac{1}{2}\right| \leq \varepsilon\right] \geq 1-\delta$$

Since $\bar{Z}$ is within ε of p and within ε of $\frac{1}{2}$, it follows that p must be within 2ε of $\frac{1}{2}$. For ε small enough, this can be simplified to:

$$\frac{1}{2} - \varepsilon \leq p \leq \frac{1}{2} + \varepsilon$$

Thus, we have shown that assuming $n = O\left(\frac{1}{\varepsilon^2}\log\left(\frac{1}{\delta}\right)\right)$, we can ensure that:

$$\frac{1}{2} - \varepsilon \leq \mathbf{Pr}[Y \leq m] = p \leq \frac{1}{2} + \varepsilon$$

with probability at least $1-\delta$.

2. Given that $E[X] = \mu$ and $stddev[X] = \sigma > 0$, define the standardized variable $Z = \frac{X-\mu}{\sigma}$. Then, we WTS

$$Pr[X \geq \mu + t\sigma] = Pr[Z \geq t] \leq \frac{1}{t^2+1}$$

Since $E[X] = 0$, and $Var[Z] = 1$, then $E[Z^2] = 1$. And given that $\frac{(Z+1/t)^2}{(t+1/t)^2} \geq 1_{\{Z \geq t\}}$, then

$$Pr[Z \geq t] = E[1_{\{Z \geq t\}}] \leq E[\frac{(Z+1/t)^2}{(t+1/t)^2}] \leq \frac{E[Z^2] + (2/t)E[Z] + 1/t^2}{(t+1/t)^2} = \frac{1+1/t^2}{(t+1/t)^2} = \frac{1}{1+t^2}$$

Thus,

$$Pr[X \geq \mu + t\sigma] \leq \frac{1}{t^2+1}$$

3. (a) Since $\|\vec{w}_i\| = 1$ for $i \in [m]$ and $\theta_{ij} = \angle(\vec{w}_i, \vec{w}_j)$, then

$$\cos(\theta_{ij}) = \frac{\vec{w}_i \cdot \vec{w}_j}{\|\vec{w}_i\|\|\vec{w}_j\|} = \vec{w}_i \cdot \vec{w}_j = \frac{\vec{v}_i \cdot \vec{v}_j}{n}.$$

Since every coordinate of each $\vec{v}_i$ is chosen to be ± 1 with probability $1/2$ each, the corresponding dimension for θ_{ij} has probability $1/2$ to be ± 1. Define each dimension to be z_i. Let $Z = \sum_{i=1}^{n} z_i$. Then,

$$\cos(\theta_{ij}) = \vec{w}_i \cdot \vec{w}_j = \frac{1}{n}\sum_{i=1}^{n} z_i.$$

Since for $i \in [n]$, $-1 \leq z_i \leq 1$, and $E[z_i] = 0 \, \forall \, i \in [n] \implies E[Z] = 0$, then by the Hoeffding Bound,

$$\mathbf{Pr}[Z \geq \delta n] \leq \exp\left(-\frac{2(\delta n)^2}{4n}\right) = \exp\left(-\frac{1}{2}\delta^2 n\right).$$

$$\begin{aligned}
\implies \mathbf{Pr}[|\cos(\theta_{ij})| \geq \delta] &= 2\,\mathbf{Pr}[\cos(\theta_{ij}) \geq \delta] \\
&= 2\,\mathbf{Pr}\left[\frac{1}{n}Z \geq \delta\right] \\
&= 2\,\mathbf{Pr}[Z \geq \delta n] \\
&\leq 2\exp\left(-\frac{1}{2}\delta^2 n\right) \\
&= \exp\left(-\frac{1}{2}\delta^2 n + \ln(2)\right) \\
&= \exp(-\Omega(\delta^2 n)).
\end{aligned}$$

Since

$$\cos\left(\frac{\pi}{2} - x\right) \approx x \text{ as } x \to 0^+,$$

the probability bound on $\cos\theta_{ij}$ translates to the angle:

$$\mathbf{Pr}\left[\left|\theta_{ij} - \frac{\pi}{2}\right| \geq \delta\right] \leq \exp(-\Omega(\delta^2 n)).$$

(b) We need to show that for some $m = \exp(\Omega(\delta^2 n))$, we have:

$$\mathbf{Pr}\left[\frac{\pi}{2} - \delta \leq \theta_{ij} \leq \frac{\pi}{2} + \delta \text{ for all pairs } i \neq j\right] \geq 0.99$$

There are $\binom{m}{2} = \frac{m(m-1)}{2}$ pairs (i, j). Using the union bound, then

$$\mathbf{Pr}\left[\bigcup_{i \neq j} |\cos\theta_{ij}| \geq \delta\right] \leq \sum_{i \neq j} \mathbf{Pr}[|\cos\theta_{ij}| \geq \delta] \leq \binom{m}{2}\exp\left(-\Omega(\delta^2 n)\right) \leq \frac{m^2}{2}\exp\left(-\Omega(\delta^2 n)\right)$$

For the probability to be at least 0.99, we need:

$$\frac{m^2}{2}\exp\left(-\Omega(\delta^2 n)\right) \le 0.01$$
$$m^2 \le 0.02\exp\left(\Omega(\delta^2 n)\right)$$
$$m \le \sqrt{0.02}\exp\left(\Omega(\delta^2 n)/2\right)$$

Take $m = 0.1\exp\left(\Omega(\delta^2 n)/2\right)$, then

$$\mathbf{Pr}\left[\bigcup_{i\neq j} |\cos\theta_{ij}| \ge \delta\right] \le \frac{m^2}{2}\exp\left(-\Omega(\delta^2 n)\right) = \frac{0.01\exp\left(\Omega(\delta^2 n)\right)}{2}\exp\left(-\Omega(\delta^2 n)\right) = 0.01/2 < 0.01$$

$$\implies \mathbf{Pr}\left[\frac{\pi}{2} - \delta \le \theta_{ij} \le \frac{\pi}{2} + \delta \text{ for all pairs } i \neq j\right] \ge 0.99$$

4. (a) Suppose X only takes on $\{t_1 < t_2 < \cdots < t_n\}$, and for $i \in [n]$, $Pr[X = t_i] = p_i$, then

$$\begin{aligned}
\int_0^\infty Pr[X \ge t]dt &= \int_0^\infty \sum_{z\ge t} Pr[X = z]dt \\
&= \int_0^\infty \sum_{i\ge t} p_i dt \\
&= (\int_0^{t_1} p_1 + p_2 + \cdots + p_n dz) + (\int_{t_1}^{t_2} p_2 + \cdots + p_n dz) + \cdots + (\int_{t_{n-1}}^{t_n} p_n dz) \\
&= (t_1 - 0)(p_1 + p_2 + \cdots + p_n) + (t_2 - t_1)(p_2 + \cdots + p_n) + \cdots + (t_n - t_{n-1})(p_n) \\
&= p_1 \cdot (t_1) + p_2 \cdot ((t_1 - 0) + (t_2 - t_1)) + \cdots + (t_n) \cdot ((t_1 - 0) + (t_2 - t_1) + \cdots + (t_n - t_{n-1}) \\
&= \sum_{i=1}^n t_i \cdot p_i \\
&= E[X]
\end{aligned}$$

   (b) With the same setting, $E[X^2] = \sum_{i=1}^n t_i^2 \cdot p_i$, then

$$\begin{aligned}
2\int_0^\infty t\,\mathbf{Pr}[X \ge t]\,dt &= 2\int_0^\infty t \sum_{t_i \ge t} p_i\,dt \\
&= 2\sum_{i=1}^n p_i \int_0^{t_i} t\,dt \\
&= 2\sum_{i=1}^n p_i(\frac{t_i^2}{2} - 0) \\
&= \sum_{i=1}^n t_i^2 \cdot p_i
\end{aligned}$$

5. (a) Since $X \ge 0$, then,

$$\begin{aligned}
\mathbf{Pr}[X = 0] &= \mathbf{Pr}[X \le 0] \\
&\le \mathbf{Pr}[|X - \mu| \ge \mu] \\
&= \mathbf{Pr}[|X - \mu| \ge \frac{\mu}{\sigma}\sigma] \\
&\le \frac{\mathrm{Var}[X]}{E[X]^2}
\end{aligned}$$

(b)

$$E[X] = E[X\mathbf{1}_{X=0}]+E[X\mathbf{1}_{X>0}] = 0+E[X\mathbf{1}_{X>0}] = E[X\mathbf{1}_{X>0}] \leq \sqrt{E[X^2]}\sqrt{E[(\mathbf{1}_{X>0})^2]} = \sqrt{E[X^2]\,\mathbf{Pr}[X>0]}$$

Hence, $\mathbf{Pr}[X > 0] \geq \frac{(E[X])^2}{E[X^2]}$

(c) If $p = o(n^{-2/3})$, then say $p = cn^{-2/3}$, where $c \to 0$ as $n \to \infty$. Define X to be the number of 4-cliques in G, then

$$E[X] = \binom{n}{4}p^6 \sim \frac{n^4}{24}p^6 = \frac{n^4c^6}{24n^4} = \frac{c^6}{24} = o(1)$$

And by Markov inequality,

$$\mathbf{Pr}[G \text{ contains a 4-clique}] = \mathbf{Pr}[X \geq 1] = E[X] = o(1)$$

(d) Suppose $p = \omega(n^{-2/3})$, then say $p = dn^{-2/3}$, where $d \to \infty$ as $n \to \infty$. Let X_i be the indicator of the $\binom{n}{4}$ possible 4-cliques, and let $X = \sum_i X_i$, then

$$E[X] = \binom{n}{4}p^6 \sim \frac{n^4}{24}p^6$$

and

$$\begin{aligned}
\implies \text{Var}[X] &= \sum_i \text{Var}(X_i) + \sum_{i\neq j} \text{Cov}(X_i, X_j) \\
&= \binom{n}{4}(E[X_i^2] - (E[X_i])^2) + \binom{n}{6}\binom{6}{2}p^{11} + \binom{n}{5}\binom{5}{3}p^9 \\
&\sim n^4p^6 + n^6p^{11} + n^5p^9 \\
\implies \frac{\text{Var}[X]}{(E[X])^2} &= \frac{n^4p^6 + n^6p^{11} + n^5p^9}{n^8p^12} = O\left(\frac{1}{n^4p^6} + \frac{1}{n^2p^1} + \frac{1}{n^3p^3}\right) = o(1)
\end{aligned}$$

Then from part (a),

$$\mathbf{Pr}[X = 0] \leq \frac{\text{Var}[X]}{(E[X])^2} = o(1)$$

Thus,

$$\mathbf{Pr}[\text{G doesn't contain a 4-clique}] = o(1)$$

4 Abstract

I'm going to be discussing linearity of expectations and how it can be used with the probabilistic method. I'll be using linearity of expectations to solve the following problems: *there is a tournament on n vertices that has at least* $\frac{n!}{2^{n-1}}$ *Hamiltonian paths*; *Any graph with m edges contains a bipartite subgraph with at least* $\frac{m}{2}$ *edges.*

5 Methods of List Decoding for 2-Deletion

5.1 Setup: The 2-Deletion Paper and the Idea of Runs

In a recent paper by Guruswami and Håstad, a novel method was introduced for constructing codes that can correct two deletions in a binary string. This method relies on a small amount of additional information, known as redundancy, to ensure that the original string can be identified from a list of size 2 after two deletions. The key to this method lies in the use of special functions, $f_1^r(x)$ and $f_2^r(x)$, which are based on the concept of "runs" in a binary string. A run is defined as a sequence of consecutive 0's or 1's in the string.

To explain these functions, consider a binary string x. The following "sketch" functions are defined:

$$f_1(x) = \sum_{i=1}^{n} i \cdot x_i \tag{1}$$

$$f_2(x) = \sum_{i=1}^{n} \binom{i}{2} \cdot x_i \tag{2}$$

$$f_1^r(x) = \sum_{i=1}^{n+1} r_i \tag{3}$$

$$f_2^r(x) = \sum_{i=1}^{n+1} \binom{r_i}{2} \tag{4}$$

To ensure that each string has a unique run pattern, the authors add a '0' at the beginning and a '1' at the end of the string. This adjustment guarantees the uniqueness of the rank sequence. For instance, the string "0011" is transformed into "(0)0011(1)", resulting in a distinct rank sequence. The functions $f_1^r(x)$ and $f_2^r(x)$ are derived from these rank sequences and play a crucial role in recovering the original string after deletions.

5.2 Code Implementation and Process

To implement this approach, we developed a code that processes strings of length n. The process involves the following steps:

- **Generating Strings:** For a given n, we first generate all binary strings of length $n-2$ (denoted as y). This is because we are focusing on correcting two deletions.
- **Generating Superstrings:** For each y, we generate all possible superstrings of length n by applying 2-insertions. These superstrings are denoted as x'.
- **Computing Sketch Values:** We then compute the sketch functions $f_1(x')$ and $f_2(x')$ for each x'.
- **Checking for Collisions:** We check how many x' strings share the same sketch values. Our goal is to ensure that no more than two strings share the same values. If more than two strings do, we consider this a collision.
- **Storing Results:** We store collisions and analyze them to identify those with the maximum list size.

Once all y strings are processed, we combine the results to identify and analyze the collisions with the maximum list size.

5.3 Five Methods on List Decoding Two Deletions

Building on the foundation provided by Guruswami and Håstad, we explored five additional methods through brute-force experiments. These experiments yielded promising results, which could potentially enhance the performance of deletion-correcting codes.

5.3.1 Method 1: Using $f_1(x)$ and $f_2(x)$ Without Runs

In this method, we attempted to use the functions $f_1(x)$ and $f_2(x)$ as defined in the original paper, instead of $f_1^r(x)$ and $f_2^r(x)$. However, when $n = 12$, the code began producing a list of size 3. For example, the strings '110001110001', '000110110001', and '000111000110', all derived from the string '0001110001', shared the same values with $f_1 = 36$, $f_2 = 131$, and the number of new runs equal to 2. This indicated that the method was unsuccessful in limiting the list size to 2.

5.3.2 Method 2: Modulo p Application on $f_1(x)$ and $f_2(x)$

This method aimed to apply modulo p to $f_1(x)$ and $f_2(x)$. However, given that Method 1 failed, applying modulo p would only increase or maintain the collision rate, leading us to conclude that Method 2 is also ineffective.

5.3.3 Method 3: Modulo p with Runs and Additional Features

Returning to the original approach using $f_1^r(x)$ and $f_2^r(x)$, we aimed to reduce redundancy. We selected the first prime p between $1.5n$ and $2n$, and computed $f_1^r(x) \mod p$ and $f_2^r(x) \mod p$. Additionally, we incorporated the counts of '0', '1', '01', and '10' in each candidate x' as part of the sketch. Our experiments demonstrated list size 2 up to $n = 20$.

While the method initially appeared successful, a more detailed analysis showed new collisions at $n = 18$ and $n = 19$, with 178 and 1684 new collisions respectively. This suggests that for larger n, further collisions might lead to a list size of 3, indicating potential limitations of this method.

However, we also introduced a new definition of randomness for binary strings, arguing that if all strings in the code are random, we can still achieve a list size of 2. We define a binary string as random if it does not contain consecutive '0's, '1's, or alternating '01's or '10's of length $\lceil \log(n) \rceil$. Applying this randomness test, we found that all new collisions were non-random, while the original collisions included random strings. For example, for n=18, one of the new collisions is on '0000000101010110000' and '0100000000101100010' generated from y='00000000010110000'. They all share the same values with $f_1^r = 18$, $f_2^r = 22$, #NewRuns = 4, #0 = 14, #1 = 5, #01 = 4, #10 = 4, and there's at least seven consecutive 0's in both strings.

5.3.4 Method 4: Using $f_1(x)$ and $f_2(x)$ Without Runs and Modulo p

In this method, we reverted to using $f_1(x)$ and $f_2(x)$ without the runs and without modulo p. This approach resulted in a list size of 3 at $n = 14$. For instance, the x-primes '01000110011101', '01000101110011', and '00110001011101' derived from the string '010001011101' all shared the same values with $f_1 = 62$, $f_2 = 294$, #NewRuns = 0, #0 = 7, #1 = 7, #01 = 4, #10 = 3.

5.3.5 Method 5: Introducing Switch String-Based Sketches

In this method, we introduced new sketch functions based on the idea of switching bits between 0 and 1. Every time the bit switches, we store the corresponding index in a switch string, using 0 for all other positions. To ensure uniqueness, we add a '0' at the beginning and a '1' at the end of each string. For example, the string $x = 010001$ becomes (0)010001(1), leading to a switch string $s = 00230060$, where the bit switches at indices 2, 3, and 6.

We then defined the following functions:

$$f_1^s(x) = \sum_{i=1}^{n+1} s_i \tag{5}$$

$$f_2^s(x) = \sum_{i=1}^{n+1} \binom{s_i}{2} \tag{6}$$

These functions were used as new sketches. Testing up to $n = 19$, we consistently observed a list size of 2. Additionally, the sizes of $f_1^s(x)$ and $f_2^s(x)$ are n and n^2, respectively, matching the original functions $f_1^r(x)$ and $f_2^r(x)$.

5.4 Sketch Function Continuations

Based on the paper by Sima and Bruck, they look for patterns of 0's and 1's defined by their synchronization vectors of length 3k, where k is the number of deletions. Based on the sketch functions we have and how they relate to patterns of 0's and 1's, the hope would be to find some relationship that lowers the redundancy further centered around saving information like sketch functions or synchronization functions (patterns of 0's and 1's) defined in the Sima and Bruck paper.

5.5 3 Deletion Case

We also took some time to test if the ideas presented by Guruswami and Håstad would translate into cases of 3 deletions. We included a new sketch defined to be: $f_3^r(x) = \sum_{i=1}^{n+1} \binom{r_i}{3}$ with the hope that the addition of this sketch along with the previous ones would give us some list size (hopefully 3 or 4) that would prove to be useful. Unfortunately, after some coding experimentation it seemed like there was a missing piece because we could not get down to a constant list size with our redundancy.