0.1 Notes on Checksum Communication Complexity

A paper has shown that the Number on Forehead communication complexity of CheckSum has been bounded as follows:

$$(\log N)^{\Omega(1)} \leq NOF(CheckSum) \leq O(\sqrt{\log N})$$

There are many different techniques to analyze something called Ramsey type Problems, where we try to maximize the size of a set while avoiding a certain pattern (like 3-Arithmetic Progressions).

- Graph Theory
- Ergodic Theory
- Fourier Analysis (Most Success)
- Polynomial Method

1 3AP Over Finite Field

The 3AP problem is the same as the original problem over $[N]$, however, we focus on the case where arithmetic is done over $\mathbb{R}_3^n$. This argument generalizes to any group where there exists a plus operation and we make a set such that:

$$\nexists a, b, c | a \neq b \neq c, a + c = 2b$$

We want to determine, given the universe $U = \{0, 1, 2\}^n$ where addition is mod 3:

$$r_3(\mathbb{F}_3^n) = \max |S| \text{ where } S \subseteq \{0, 1, 2\}^n \text{ and no 3AP exists.}$$

This is also called the "CAP-SET PROBLEM".

1.1 Size of CAP-SET

Recall that Beherend's construction led to a subset $S \subseteq [N]$, where:

$$|S| \subseteq \frac{N}{2^{c \cdot \sqrt{\log N}}}$$

In CAPSET case, the size of N is 3^n. We argue that it is easier to form a $3AP$ in $\mathbb{F}_3^n$ than in $[3^n]$.

Intuition: Finding solutions to $x + y = 2z$ modular arithmetic is easier over $\mathbb{F}_3^n$ than $\mathbb{Z}$. (I do not get his argument)

Theorem 1.1. *$r_3(\mathbb{F}_3^n) \leq (2.76)^n = N^c$ for some $c < 1$.*

Recall that $N/(2^{c\cdot\sqrt{\log N}}) \leq r_3(\mathbb{Z}_n)$

1.2 Examples and Properties

Firstly, we know that we can select a set that has size 2^n that satisfies that there are no $3AP$'s. We argue that $A = \{0,1\}^n$ has no $3AP$'s.

Proof. Assume that there was some a, b, c such that they form a nontrivial $3AP$. The argument is relatively straightforward, but we know that if $a_i + b_i$ is 1, c_i cannot be a three AP, if $a_i + b_i = 0$ then $a_i = b_i = c_i = 0$ so they must be the same bit and if $a_i + b_i = 2$ then $c_i = b_i = a_i = 1$. So either the three numbers are the same or they do not form a $3AP$. □

This provides a lower bound for $r_3(\mathbb{F}_3^n)$:

$$r_3(\mathbb{F}_3^n) \geq 2^n = N^{\log_3 2} \approx N^{0.63}$$

What do $3AP$s over mod 3 arithmetic look like? We know that:

$$(a, b, c) \in \{0,1,2\}^n \text{ and } a + b = 2c$$

We can think of it as:

$$\begin{aligned} a + b &\equiv 2c \mod 3 \\ \forall i, a_i + b_i &\equiv 2c_i \mod 3 \\ a_i + b_i &\equiv -c_i \mod 3 \\ a_i + b_i + c_i &\equiv 0 \mod 3 \end{aligned}$$

This equation is satisifed when all $a_i = b_i = c_i$ or each value is unique.

1.3 Lower Bound on $r_3(\mathbb{F}_3^n)$

Theorem 1.2. *$r_3(\mathbb{F}_3^n) \leq (2.76)^n = N^c$ for some $c < 1$.*

Proof Outline: We are going to use the polynomial method which will aim to do the following:

- Encode objects or desired solutions as polynomials
- Use linear algebra (rank / dimension)

- Use interpolation
- Fast polynomial evaluation

Generally we will show that if there exists some large 3AP free set, it implies that some property of identity tensor is violated.

Definition 1.1. We say that a **Rank** of a matrix M ($rank(M)$) is the number of linearly independent columns or number of linearly independent rows of the matrix. Another way to think about it is the number of rank 1 matrices that add up to M:

$$Rank(M) = \min_r \{M = M_1 + ... + M_r\}$$

Where all M_i are rank 1 matrices.

Knowing that $rank(M) = r$, we can write M as the following:

$$M = f_1 g_1^T + ... + f_r g_r^T$$

Where each f_i represents one of the linearly independent columns and each g_i represents one of the linearly independent rows. This decomposition has each $M_i = f_i \cdot g_i^T$. We can therefore redefine rank for matrices in $\mathbb{F}_3^{n \times n}$ as the following:

$$\begin{aligned} Rank(M) = \min_r \Big\{ & \exists f_1, \ldots, f_r : [N] \longrightarrow \mathbb{F}_3, \\ & \exists g_1, \ldots, g_r : [N] \longrightarrow \mathbb{F}_3 \\ & \mid M(x,y) = f_1(x)g_1(y) + \cdots + f_r(x)g_r(y) \ \forall x, y \in [N] \Big\} \end{aligned}$$

We should also think of matrices as a function that takes in two inputs and outputs a value. Furthermore, for a tensor, we can think of it as a function that takes in three inputs and outputs a value. For tensor rank, the general notion is as follows:

Definition 1.2. We say that a **Rank** of a tensor T ($rank(T)$) is the number of rank 1 tensors that add up to T:

$$Rank(T) = \min_r \{T = T_1 + ... + T_r\}$$

Where all T_i are rank 1 tensors.

We draw a parallel with the second definition of matrix rank and define the following for tensors:

$$Rank(T) = \min_r \Big\{ \exists f_1, \ldots, f_r : [N] \longrightarrow \mathbb{F}_3, \\ \exists g_1, \ldots, g_r : [N] \longrightarrow \mathbb{F}_3, \\ \exists h_1, \ldots, h_r : [N] \longrightarrow \mathbb{F}_3, \\ \mid T(x,y,z) = f_1(x)g_1(y)h_1(z) + \cdots + f_r(x)g_r(y)h_r(z) \ \forall x,y,z \in [N] \Big\}$$

You think of a rank one tensor as a function that can be written as a product of three functions, each of which takes in one of the three inputs. We will be interested in the following tensor:

$$\mathbb{I}(x,y,z) = \begin{cases} 1 & \text{if } x = y = z \\ 0 & \text{otherwise} \end{cases}$$

Firstly, we know that $rank(\mathbb{I}^{N \times N}) \leq N$ because we can write $\mathbb{I}$ as the sum of N rank one tensors. Secondly, we know that $rank(\mathbb{I}^{N \times N \times N}) \leq N$ where $f_i(x) = g_i(y) = h_i(z) = 1$ if $x = y = z = i$ and 0 otherwise. For each f_i, g_i, h_i we just say $f_i(x) = 1$ if $x = i$ and 0 otherwise.

1.3.1 Thought Experiment

Say we have a matrix M which we think about as a function $M(x,y)$. If two parties are trying to figure out the value of $M[x,y]$ where one party has x and the other has y, one easy protocol is having the first party send x to the second party and having the second party compute $M[x,y]$. However, if M has low rank, we can do better. We can write M as the sum of r rank one matrices:

$$M = f_1 g_1^T + ... + f_r g_r^T$$

In this case, we can have Alice (party 1) send $f_i(x)$ for all i to Bob. Since $f_i(x)$ is either $[0,1,2]$, the complexity of this protocol is just $O(r)$ opposed to $O(\log N)$.

Consider the case with three parties trying to compute the tensor $M^{n \times n \times n}$. If M has low rank, we can write M as the sum of r rank one tensors:

$$M = f_1 g_1 h_1 + ... + f_r g_r h_r$$

In this case, we can have Alice and Bob send $f_i(x)$ and $g_i(y)$ for all i to Charlie, and Charlie can compute $M[x,y,z]$ by computing $f_i(x)g_i(y)h_i(z)$ for all i. The complexity of this protocol is just $O(r)$ opposed to $O(\log N)$ once more.

1.3.2 New Notion of Tensor Rank

We define a new notion of tensor rank called Slice Rank. To motivate this, we make a continuation of the previous thought experiment where each round, two players can get together and send information to the third player. We define the protocol as:

- Alice has x, Bob has y, and Charlie has z.
- Start with Alice and Bob in a room and Charlie in another room. Alice and Bob can talk to each other but not to Charlie.
- In other rounds, any two players can get together and talk to each other but not to the third player.

To formalize this idea, we define the following:

Definition 1.3. We say that a **Slice Rank** of a tensor T ($srank(T)$)

$$\text{SRank}(T) = \min \left\{ r : \begin{array}{l} \exists\ r_1, r_2, r_3 \text{ with } r_1 + r_2 + r_3 = r, \\ f_1, \ldots, f_{r_1} : [M] \to \mathbb{F}, \\ \widetilde{f}_1, \ldots, \widetilde{f}_{r_1} : [M] \times [M] \to \mathbb{F}, \\ g_1, \ldots, g_{r_2} : [M] \to \mathbb{F}, \\ \widetilde{g}_1, \ldots, \widetilde{g}_{r_2} : [M] \times [M] \to \mathbb{F}, \\ h_1, \ldots, h_{r_3} : [M] \to \mathbb{F}, \\ \widetilde{h}_1, \ldots, \widetilde{h}_{r_3} : [M] \times [M] \to \mathbb{F}, \\ \text{such that} \\ T(x,y,z) = \displaystyle\sum_{i=1}^{r_1} f_i(x)\widetilde{f}_i(y,z) + \sum_{i=1}^{r_2} g_i(y)\widetilde{g}_i(x,z) \\ \qquad\qquad + \displaystyle\sum_{i=1}^{r_3} h_i(z)\widetilde{h}_i(x,y) \end{array} \right\}.$$

We end this lecture by explaining that $SRank(T) \leq Rank(T)$ since we can just define $\widetilde{f}_i$ as $g_i \cdot h_i$.