Sum-Check

May 29, 2025

Flying Nobita | ProgCryptoSG

Contents

Sum-check Overview

Algebraic Methods for Interactive Proof Systems (1992)

Authors: Carsten Lund, Lance Fortnow, Howard Karloff, Noam Nisan

- one of the most important interactive proof in the ZK literature
- allows the prover to convice the verifier that a multivariate polynomial sum over the Boolean Hypercube with a claimed sum C_1. i.e.:
  - $C_1 \stackrel{?}{=} H := \sum_{b_1 \ldots b_v \in \{0,1\}} g(b_1, \ldots, b_v)$
    - Where:
      - C_1 is the claimed sum
      - H is the true sum
      - g is a v-variate polynomial defined over a finite field $\mathbb{F}$

Multivariate Polynomial

A multivariate polynomial that is non-linear in each of the variables

$P(x1, x2, x3) = x_1{x_2}^2 + 3x_1x_2 + 4{x_2}^5$

- $TotalDeg(P) =$ maximum degree of all terms
  - e.g. 5 because of ${4x_2}^5$

Multilinear (Special Case)

- A multivariate polynomial that is linear in each of its variables separately
- e.g. $P(x, y, z) = 3 + 4x + 5y + 5xy + z$

Boolean Hypercube

- The set of points $\{0,1\}^n$
- e.g. for $n = 3$, it would be:

$$
\begin{aligned}
&(0,0,0), && (1)\\
&(0,0,1), && (2)\\
&(0,1,0), && (3)\\
&(0,1,1), && (4)\\
&(1,0,0), && (5)\\
&(1,0,1), && (6)\\
&(1,1,0), && (7)\\
&(1,1,1) && (8)
\end{aligned}
$$

Protocol

Round 0

Prover calculates and send the claimed sum to the verifier.

Prover:

- send value C_1
- claimed $C_1 = H$

EXAMPLE R0

$$C_1 \stackrel{?}{=} H := \sum_{x_1, x_2, x_3 \in \{0,1\}} g(x_1, x_2, x_3)$$

$$\begin{aligned} \text{Let } g(X_1, X_2, X_3) &= 2{X_1}^3 + X_1 X_3 + X_2 X_3 \\ H &= g(0,0,0) + \ldots + g(1,1,1) \\ H &= 1 + 2 + 3 + 2 + 4 \\ H &= 12 \end{aligned}$$

$$\text{Honest prover send: } C_1 = 12$$

Protocol - R1 - Prover

The prover computes g over the Boolean Hypercube except leaving the 1st variable X_1 open. Then let the Verifier evaluate $g_1(X_1)$.

1. Prover:

- sends $g_1(X_1)$, which is a *univariate polynomial*
- claims $g_1(X_1) = s_1(X_1)$, where:
  - $s_1(X_1) := \sum_{(x_2,\ldots,x_v)\in\{0,1\}^{v-1}} g(X_1, x_2, \ldots, x_v)$
  - $s_1(X_1)$ is also defined as $H = s_1(0) + s_1(1)$
  - i.e. $s_1(X_1)$ **is the summation of g over Boolean Hypercube for all variables except x_1 which is equal to X_1**

EXAMPLE R1 - PROVER

Prover claims:

$$
\begin{aligned}
g_1(X_1) &= s_1(X_1) \\
s_1(X_1) &= g(X_1, 0, 0) + g(X_1, 0, 1) \\
&\quad + g(X_1, 1, 0) + g(X_1, 1, 1) \\
&= 8{X_1}^3 + 2X_1 + 1
\end{aligned}
$$

Protocol - R1 - Verifier

Verifier checks Prover's claim and send r_1 to Prover

Verifier:

checks:

1. $C_1 \stackrel{?}{=} H = g_1(0) + g_1(1)$

If true, then while Prover verified C_1 is tied to g_1,

Verifier still need to check $g_1(X_1) = s_1(X_1)$

Verifier pick random r_1 and check $g_1(r_1) = s_1(r_1)$

- $g_1(r_1)$ is explicit
  - is *easy* to compute
- $s_1(r_1)$ is summed over the hypercube
  - is *hard* to compute
- but **Verifier can use sum-check to check** $s_1(r_1)$

EXAMPLE R1 - VERIFIER

Prover already sent $C_1 = 12$.

Verifier compute and check:

$$\begin{aligned} s_1(X_1) &= 8{X_1}^3 + 2X_1 + 1 \\ &= s_1(0) + s_1(1) \\ &= 12 \\ &= C_1 \end{aligned}$$

Verifier pick $r_1 = 2$ and send to Prover.

Protocol - R2 to v - 1

The Prover replaces X from the previous round with random numbers given by the Verifier. The Prover leaves the current X open and computes g with the rest of X. The Verifier evaluates the current X.

Prover:

- sends $g_j(X_j)$
- claims $g_j(X_j) = s_j(X_j)$
  - where $s_j(X_j) :=$

$$\sum_{(x_{j+1},\ldots,x_v)\in\{0,1\}^{v-j}} g(r_1,\ldots,r_{j-1},X_j,x_{j+1},\ldots,x_v)$$

Verifier:

- checks:

$$s_{j-1}(r_{j-1}) \stackrel{?}{=} g_j(0) + g_j(1)$$

- sends a random element $r_j \in \mathbb{F}$

EXAMPLE - R2

Honest prover prepare and send:

$$\begin{aligned} g_2(x_2) &= g(2, X_2, 0) + g(2, X_2, 1) \\ &= 34 + X_2 \end{aligned}$$

Verifier checks:

$$s_1(r_1) \stackrel{?}{=} g_2(0) + g_2(1), \text{where } r_1 = 2$$

$$s_1(X_1) = 8{X_1}^3 + 2X_1 + 1$$

$$s_1(2) = 69$$

$$g_2(0) + g_2(1) = 34 + 35 = 69$$

Verifier pick $r_2 = 3$ and send to Prover.

Protocol - Last Round

Prover:

send $g_v(X_v)$

claim $g_v(X_v) = s_v(X_v)$

- where $s_v(X_v) := g(r_1, \ldots, r_{v-1}, X_v)$

Verifier:

checks:

$g_{v-1}(r_{v-1}) \stackrel{?}{=} g_v(0) + g_v(1)$

select a random element $r_v \in \mathbb{F}$

checks $g_v(r_v) \stackrel{?}{=} g(r_1, \ldots, r_v)$ via any of:

oracle access to g

compute $g(r_1, \ldots, r_v)$ on its own

ask Prover whom can prove the claim via running further sum-checks

EXAMPLE - R3

Honest Prover prepare and send:

$$\begin{aligned} g_3(X_3) &= g(2, 3, X_3) \\ &= 16 + 5X_3 \end{aligned}$$

Verifier checks:

$$\begin{aligned} s_2(r_2) &\stackrel{?}{=} g_3(0) + g_3(1), \text{where } r_2 = 3 \\ s_2(X_2) &= 34 + X_2 \\ s_2(3) &= 37 \\ g_3(0) + g_3(1) &= 16 + 21 = 37 \end{aligned}$$

Verifier pick $r_3 = 6$ and check:

$$\begin{aligned} s_3(6) &\stackrel{?}{=} g(2, 3, 6) \\ s_3(6) &= 16 + 30 = 46 \\ g(2, 3, 6) &= 16 + 12 + 18 = 46 \text{ (e.g. via orcale)} \end{aligned}$$

Security

COMPLETENESS ERROR

- $\delta_c = 0$

If the Prover is honest, then the Verifier will always accept when following the protocol.

SOUNDNESS ERROR

- $\delta_S \leq \frac{v\ deg(g)}{|\mathbb{F}|}$

Proof is based on the *Schwartz-Zippel lemma*.
i.e. 2 different polynomials almost never match at randomly chosen points

Applications

INTERACTIVE PROOF

- GKR08

SNARKS

- Spartan20
- Brakedown21
- Orion22
- HyperPlonk23

LOOKUPS

- Locq24

Take Aways

- $\sum_{b_1 \ldots b_v \in \{0,1\}} g(b_1, \ldots, b_v)$
  - number of evaluations: 2^v
  - time to compute each evaluation: T
  - total time to compute **without sum-check**: $2^v \cdot T$
  - total time to compute **with sum-check**: ~T
- **Prover's perspective:** allows the Prover to prove to the Verifier that the sum is correct
- **Verifier's perspective:** allows the Verifier to reduce its computation when calculating the sum by trustlessly delegating the computation to a Prover
- a fundamental building block for some of the **fastest** prover amongst interactive proof system

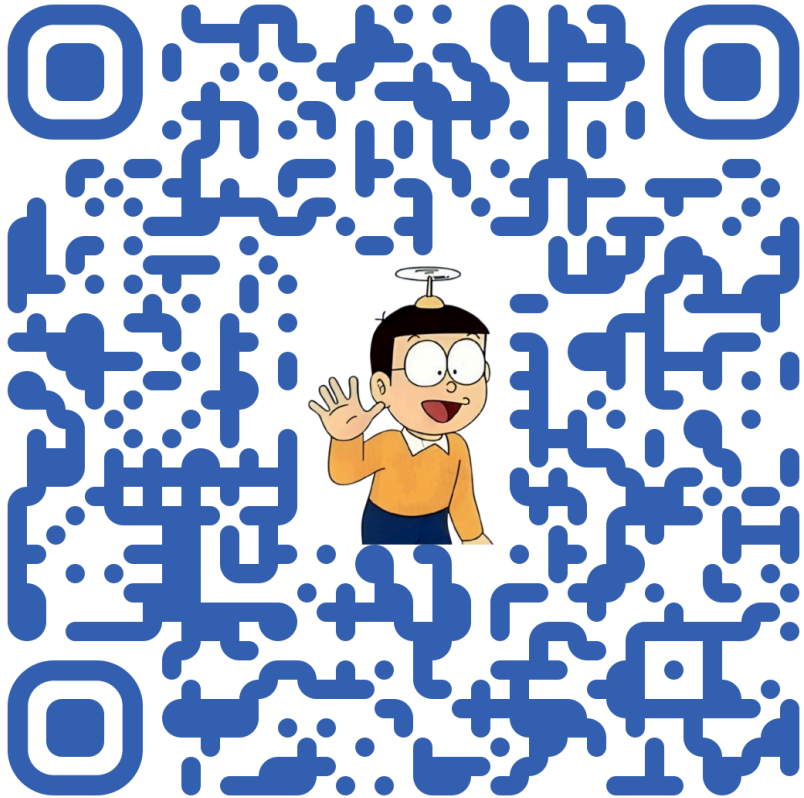
Learn More

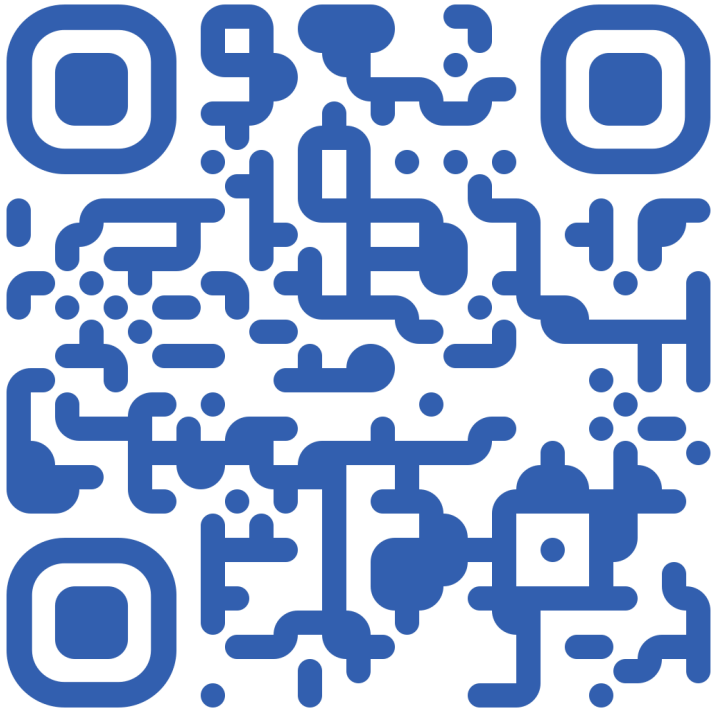
REFERENCES

- Proofs, Argument & Zero Knowledge by Justin Thaler
- GKR and Sumcheck Protocol - Yupeng Zhang

OTHER RESOURCE

Sum-Check Protocol - FlyingNobta.com





1 min Anonymous Feedback

Thank you!

𝕏 @FlyingNobita