A Summary of the Sumcheck Protocol

Lizheng Wang 22nd September 2025
lizhengwang1124@gmail.com

1 Introduction to the Sumcheck Protocol

The Sumcheck Protocol is a fundamental and powerful interactive proof system used in theoretical computer science and, more recently, as a cornerstone of many Zero-Knowledge Proof (ZKP) systems. Its primary purpose is to allow a Prover (P) to convince a Verifier (V) of the value of a large sum, without the Verifier needing to perform the sum themselves. The author believes that sumcheck is the core component of SNARK to get succinctness.

The problem is as follows: Given a field $\mathbb{F}$ and a low-degree multivariate polynomial $f(X_1, \ldots, X_m)$ over $\mathbb{F}$, the Prover claims to know the value H such that:

$$H = \sum_{x_1 \in \{0,1\}} \cdots \sum_{x_m \in \{0,1\}} f(x_1, \ldots, x_m)$$

The Sumcheck protocol allows the Verifier to check this claim with high probability by interacting with the Prover for m rounds. The Verifier's work is significantly less than computing the 2^m terms of the sum, making it highly efficient.

2 The Classic Multivariate Sumcheck (Hypercube)

The original protocol, introduced by Lund, Fortnow, Karloff, and Nisan[1], operates over the boolean hypercube $\{0,1\}^m$.

2.1 The Protocol Steps

The protocol proceeds in m rounds, one for each variable X_j.

1. **Round 1 (Variable X_1):**
   - The Prover defines a new single-variable polynomial $f_1(X_1)$ by summing f over all other variables:
   $$f_1(X_1) = \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_m \in \{0,1\}} f(X_1, x_2, \ldots, x_m)$$
   - The Prover sends the polynomial $f_1(X_1)$ to the Verifier (e.g., as a list of coefficients).
   - The Verifier checks that f_1 has the expected degree.
   - The Verifier checks if $f_1(0) + f_1(1) = H$. If this check fails, the Verifier rejects.
   - The Verifier samples a random challenge $r_1 \in \mathbb{F}$ and sends it to the Prover.
   - The new claim to be proven is $H_1 = f_1(r_1)$.
2. **Round j (for $j = 2, \ldots, m$):**
   - The Prover and Verifier have a new, smaller claim:
   $$H_{j-1} = \sum_{x_j \in \{0,1\}} \cdots \sum_{x_m \in \{0,1\}} f(r_1, \ldots, r_{j-1}, x_j, \ldots, x_m)$$

- The Prover defines $f_j(X_j)$ by summing over the remaining variables:

$$f_j(X_j) = \sum_{x_{j+1}\in\{0,1\}} \cdots \sum_{x_m\in\{0,1\}} f(r_1, \ldots, r_{j-1}, X_j, x_{j+1}, \ldots, x_m)$$

- The Prover sends $f_j(X_j)$ to the Verifier.
- The Verifier checks the degree of f_j.
- The Verifier checks if $f_j(0) + f_j(1) = H_{j-1}$. If not, reject.
- The Verifier samples a random challenge $r_j \in \mathbb{F}$ and sends it to the Prover.
- The new claim is $H_j = f_j(r_j)$.

3. **Final Step (After m rounds):**

- The Verifier has received m polynomials, checked m sums, and generated a random vector $r = (r_1, \ldots, r_m)$.
- The final claim is H_m.
- The Verifier must now check if H_m is consistent with the original polynomial f at the random point r.
- The Verifier **computes** $f(r_1, \ldots, r_m)$ **themselves** and checks if $f(r_1, \ldots, r_m) = H_m$.
- If the check passes, the Verifier accepts the original claim H.

2.2 Complexity and Soundness

The Verifier's work is dominated by checking m univariate polynomials (which involves m additions) and performing one final evaluation of f. This is typically $O(m \cdot d + \text{eval}_f)$, where d is the max degree in any variable and eval_f is the cost to evaluate f. This is exponentially faster than the $O(2^m)$ cost of the naive sum. By the Schwartz-Zippel lemma, if the Prover is dishonest, they will be caught with high probability $\frac{m \cdot d}{\mathbb{F}}$.

And using the bookkeeping table and dynamic programming technique in [2], the complexity of the prover is linear to the original problem($O(2^m)$).

3 Univariate Sumcheck

In many modern ZKPs (like Marlin[3] and Aurora[4]), the goal is not to sum a multivariate polynomial f over a hypercube, but to sum a *univariate* polynomial $g(X)$ over a large domain $D \subset \mathbb{F}$.

The core idea is to find a mapping (an isomorphism) from the boolean hypercube $\{0,1\}^m$ to the domain D, where $|D| = n = 2^m$. This allows us to redefine the problem in terms of a new m-variate polynomial f, and then apply the classic sumcheck protocol from Section 2.

3.1 Sumcheck over Multiplicative Subgroups (Marlin-style)

In systems like Marlin[3], the goal is to prove a statement of the form $\sum_{x\in H} F(x) = \sigma$, where H is a multiplicative subgroup of a finite field $\mathbb{F}$ with size $|H| = n$. This protocol does not use the classic iterative structure. Instead, it converts the sum-check claim into a single polynomial identity check, which is highly efficient when used with polynomial commitment schemes. Firstly, we need to introduce a lemma.

Lemma

For any polynomial $g(X) \in \mathbb{F}_{<n}(X)$ of degree less than n, and for a multiplicative subgroup $H \subset \mathbb{F}$ of size n, the following equality holds:

$$\sum_{x \in H} g(x) = n \cdot g(0)$$

Proof

We represent the polynomial $g(X)$ in its coefficient form. Since the degree of $g(X)$ is less than n, we can write:

$$g(X) = c_0 + c_1 X + c_2 X^2 + \cdots + c_{n-1} X^{n-1}$$

Here, the constant term of the polynomial is c_0, which is equal to $g(0)$.
Now, we compute the sum of the evaluations of $g(X)$ at all points in the subgroup H:

$$\sum_{x \in H} g(x) = \sum_{x \in H} \left(\sum_{i=0}^{n-1} c_i x^i \right)$$

By swapping the order of summation, we get:

$$\sum_{x \in H} g(x) = \sum_{i=0}^{n-1} \sum_{x \in H} c_i x^i = \sum_{i=0}^{n-1} c_i \left(\sum_{x \in H} x^i \right)$$

The key to the proof lies in evaluating the inner sum, $\sum_{x \in H} x^i$. Since H is a multiplicative subgroup of order n, its elements are the n-th roots of unity in the field $\mathbb{F}$. The sum of powers of these roots of unity has the following important property:

$$\sum_{x \in H} x^i = \begin{cases} n & \text{if } i \equiv 0 \pmod{n} \\ 0 & \text{if } i \not\equiv 0 \pmod{n} \end{cases}$$

We use this property to analyze each term in our summation, where i ranges from $0, 1, \ldots, n-1$:

- **Case 1:** $i = 0$**.** The inner sum is $\sum_{x \in H} x^0 = \sum_{x \in H} 1 = |H| = n$.
- **Case 2:** $1 \leq i \leq n-1$**.** In this range, i is not a multiple of n. Therefore, the inner sum is $\sum_{x \in H} x^i = 0$.

Substituting these results back into the main equation:

$$\sum_{x \in H} g(x) = c_0 \left(\sum_{x \in H} x^0 \right) + c_1 \left(\sum_{x \in H} x^1 \right) + \cdots + c_{n-1} \left(\sum_{x \in H} x^{n-1} \right)$$

$$\sum_{x \in H} g(x) = c_0 \cdot (n) + c_1 \cdot (0) + \cdots + c_{n-1} \cdot (0)$$

$$\sum_{x \in H} g(x) = n \cdot c_0$$

Since $c_0 = g(0)$, we arrive at the final result:

$$\sum_{x \in H} g(x) = n \cdot g(0)$$

This leads to a crucial relationship: if we divide $F(X)$ by the vanishing polynomial of the subgroup, $V_H(X) = X^n - 1$, the sum is determined entirely by the remainder.

$$F(X) = h(X) \cdot V_H(X) + R(X)$$

where $\deg(R) < n$. The key identity is that the sum over the subgroup is proportional to the constant term of the remainder polynomial $R(X)$:

$$\sum_{X \in H} F(X) = n \cdot R(0)$$

Therefore, the claim $\sum_{X \in H} F(X) = \sigma$ is equivalent to the claim that the constant term of the remainder is $R(0) = \sigma/n$.

The protocol leverages this to check the sum:

1. **Prover's Computation**: The Prover and Verifier agree on the claimed sum σ. The Prover, who knows the polynomial $F(X)$ (which is typically committed to beforehand), computes the quotient $h(X)$ and remainder $R(X)$ from the division $F(X)/V_H(X)$.
2. **Polynomial Construction**: To prove that $R(0) = \sigma/n$, the Prover defines a new polynomial $g(X) = \frac{R(X)-R(0)}{x}$.[1]. By construction. The original identity can now be rewritten as:

$$F(X) = h(X) \cdot V_H(X) + x \cdot g(X) + \frac{\sigma}{n}$$

3. **Interaction**: The proof reduces to verifying this polynomial equation.
   - The Prover sends commitments to the polynomials $h(X)$ and $g(X)$ to the Verifier.
   - The Verifier generates a random challenge point $\alpha \in \mathbb{F}$.
   - The Prover provides evaluations of the polynomials at α: $F(\alpha)$, $h(\alpha)$, and $g(\alpha)$. These are provided as openings of the respective polynomial commitments.
4. **Verifier's Check**: The Verifier performs a single check at point α:

$$F(\alpha) \stackrel{?}{=} h(\alpha) \cdot V_H(\alpha) + \alpha \cdot g(\alpha) + \frac{\sigma}{n}$$

   The Verifier can compute $V_H(\alpha) = \alpha^n - 1$ itself. If the equality holds, the Verifier is convinced (with high probability, by the Schwartz-Zippel lemma) that the polynomial identity is correct, and therefore that the original sum was indeed σ.

3.2 Sumcheck over Additive Cosets (Aurora-style)

In systems like Aurora[4], the sum is performed over an additive coset of a vector subspace. Let $\mathbb{Z}_H(X)$ be the vanishing polynomial of the subspace H. Any polynomial $\hat{f}(X)$ of degree less than

[1] If we just use $R(x) - R(0)$, there is an attack. The prover want to convice $\sum_{x \in H} F(x) = \sigma'$, then he send $g'(x) = \frac{\sigma - \sigma'}{n} + g(x)$, Then $F(\alpha) = h(\alpha) \cdot V_H(\alpha) + g'(\alpha) + \frac{\sigma'}{n}$.

d can be uniquely expressed as:

$$\hat{f}(X) \equiv \hat{g}(X) + \mathbb{Z}_H(X) \cdot \hat{h}(X)$$

where $\deg(\hat{g}) < |H|$ and $\deg(\hat{h}) < d - |H|$.

Since $\mathbb{Z}_H(a)$ is zero for all $a \in H$, the sum of $\hat{f}$ over H is equivalent to the sum of $\hat{g}$ over H:

$$\sum_{a \in H} \hat{f}(a) = \sum_{a \in H} \left(\hat{g}(a) + \mathbb{Z}_H(a) \cdot \hat{h}(a) \right) = \sum_{a \in H} \hat{g}(a)$$

A theorem by Byott and Chapman provides the crucial insight: the sum $\sum_{a \in H} \hat{g}(a)$ is directly determined by the coefficient of the $X^{|H|-1}$ term in $\hat{g}(X)$. If we denote this coefficient by β, then the sum is equal to $\beta \cdot \xi$, where $\xi = \sum_{a \in H} a^{|H|-1}$ is a non-zero constant derivable from H.

This transforms the sum verification problem into a coefficient verification problem. The protocol is designed to check this property efficiently.

Protocol Specification

The non-interactive version of the protocol proceeds as follows:

1. **Prover's Computation:** The prover decomposes the polynomial $\hat{f}(X)$ to find the components $\hat{g}(X)$, $\hat{h}(X)$, and the coefficient β such that $\hat{f}(X) \equiv \hat{g}(X) + \beta X^{|H|-1} + \mathbb{Z}_H(X)\hat{h}(X)$, where $\deg(\hat{g}) < |H| - 1$.

2. **Prover's Message:** The prover sends the evaluation of $\hat{h}(X)$ over a large domain L to the verifier.

3. **Verifier's Check:** The verifier performs a single, unified check that confirms both the correctness of the decomposition and the consistency of the coefficient β with the claimed sum μ. This is done by constructing a new polynomial, $\hat{p}(X)$, and verifying that its degree is less than $|H| - 1$:

   $$\hat{p}(X) := \xi \cdot \hat{f}(X) - \mu \cdot X^{|H|-1} - \xi \cdot \mathbb{Z}_H(X)\hat{h}(X)$$

   In the IOP framework, this degree check is performed efficiently using a low-degree test on the evaluation of $\hat{p}(X)$.

Proof of Correctness

The protocol's correctness hinges on the following lemma.[2]

Lemma

Let H be an affine subspace of $\mathbb{F}$, and let $\hat{g}(X)$ be a univariate polynomial over $\mathbb{F}$ of degree strictly less than $|H| - 1$. Then:

$$\sum_{a \in H} \hat{g}(a) = 0$$

Completeness

For an honest prover, the claimed sum is correct, so $\mu = \sum \hat{f}(a)$. As shown earlier, this implies $\mu = \beta\xi$. When we substitute this into the verifier's check polynomial, $\hat{p}(X)$, the terms involving

[2] The proof is in Appendix A of [4]

$X^{|H|-1}$ cancel out:

$$\begin{aligned}\hat{p}(X) &\equiv \xi \cdot (\hat{g}(X) + \beta X^{|H|-1} + \mathbb{Z}_H(X)\hat{h}(X)) - (\beta\xi) \cdot X^{|H|-1} - \xi \cdot \mathbb{Z}_H(X)\hat{h}(X) \\ &\equiv \xi \cdot \hat{g}(X)\end{aligned}$$

Since $\deg(\hat{g}) < |H| - 1$, the degree of $\hat{p}(X)$ is also less than $|H| - 1$. The verifier's low-degree test will pass, and the proof is accepted.

Soundness

If the prover is dishonest, the true sum is $\mu' \neq \mu$. If the verifier were to accept, it would mean that the verifier's check polynomial $\hat{p}(X)$ passed a low-degree test, implying $\deg(\hat{p}) < |H| - 1$. According to the lemma, this means $\sum_{a \in H} \hat{p}(a) = 0$.

However, a direct calculation of this sum yields:

$$\begin{aligned}\sum_{a \in H} \hat{p}(a) &= \sum_{a \in H} \left(\xi \cdot \hat{f}(a) - \mu \cdot a^{|H|-1}\right) \\ &= \xi \sum_{a \in H} \hat{f}(a) - \mu \sum_{a \in H} a^{|H|-1} \\ &= \xi \cdot \mu' - \mu \cdot \xi = \xi(\mu' - \mu)\end{aligned}$$

Since $\xi \neq 0$ and $\mu' \neq \mu$, this sum is non-zero. This is a contradiction. Therefore, the polynomial $\hat{p}(X)$ cannot have a low degree, and the verifier's test will fail with high probability.

4 References

[1] Carsten Lund et al. 'Algebraic methods for interactive proof systems'. In: *Journal of the ACM (JACM)* 39.4 (1992), pp. 859–868.

[2] Justin Thaler. 'Time-optimal interactive proofs for circuit evaluation'. In: *Annual cryptology conference*. Springer. 2013, pp. 71–89.

[3] Alessandro Chiesa et al. 'Marlin: Preprocessing zkSNARKs with universal and updatable SRS'. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2020, pp. 738–768.

[4] Eli Ben-Sasson et al. 'Aurora: Transparent succinct arguments for R1CS'. In: *Annual international conference on the theory and applications of cryptographic techniques*. Springer. 2019, pp. 103–128.