Univariate sumcheck protocol

Karthik Inbasekar[a]

[a] *Ingonyama*

E-mail: karthik@ingonyama.com

Abstract: Univariate sumcheck

1 Univariant Polynomial Sumcheck

1.1 Problem statement

Consider a univariate polynomial in Lagrange basis constructed from n data points $v_i \in \mathbb{F}$ in a domain $|H| = n$

$$F(X) = \sum_{i=0}^{n-1} v_i \cdot \mathbb{L}_{x_i}(X) \tag{1.1}$$

where the Lagrange polynomials $\mathbb{L}_{x_i}$ of such that $deg(\mathbb{L}) \leq n-1$ have their usual properties. The sumcheck problem [1] is the question

$$\sum_{X \in H} F(X) \stackrel{?}{=} \sigma \tag{1.2}$$

where $\sigma \in \mathbb{F}$. Expanding the sum in the LHS of (1.2) we get

$$\sum_{X \in H} F(X) = \sum_{X \in H} \sum_{i=0}^{n-1} v_i \cdot \mathbb{L}_{x_i}(X) = \sum_{i=0}^{n-1} v_i \tag{1.3}$$

Since each of the Lagrange polynomials are non vanishing exactly at one point in the domain, for each i there exist exactly one point in the sum $X \in H$ which gives a non vanishing result. Thus the sumcheck problem (1.2) is equal to

$$\sum_{i=0}^{n-1} v_i \stackrel{?}{=} \sigma \tag{1.4}$$

We interpolate from (1.1) into a monomial basis of degree d (at this point d is arbitrary for generality)

$$f(X) = \sum_{i=0}^{d} a_i \cdot X^i \tag{1.5}$$

where $a_i \in \mathbb{F}$. The v_i in (1.1) are related to a_i via $a_i = f(x_i)\ \forall x_i \in H$. The sumcheck in the coefficient form is defined by

$$\sum_{X \in H} f(X) \stackrel{?}{=} \sigma \tag{1.6}$$

Let H be spanned by roots of unity $\{1, h, h^2, , \ldots h^{n-1}\}$, where $|H| = n$. Expanding the sum in the LHS of (1.6) we get

$$\sum_{X \in H} f(X) = f(1) + f(h) + \ldots f(h^{n-1}) = \sum_{i=0}^{n-1} v_i \tag{1.7}$$

Expanding each term in the sum (1.7) using $f(X) = \sum_{i=0}^{d} a_i X^i$ and grouping the terms in terms of the coefficients a_i we get

$$\sum_{X \in H} f(X) = a_0 \cdot |H| + a_1 \cdot \sum_{i=0}^{n-1} h^i + a_2 \cdot \sum_{i=0}^{n-1} h^{2i} + \ldots + a_d \cdot \sum_{i=0}^{n-1} h^{i \cdot d} \tag{1.8}$$

Since $h^n = 1$, some of the sums vanish due to the property

$$\sum_{i=0}^{n-1} h^{i \cdot d} = \frac{1 - h^{n \cdot d}}{1 - h^d} = \begin{cases} |H| \text{ if } d \equiv 0 \mod n \\ 0 \text{ if } d = \{1, \ldots n-1\} \end{cases} \tag{1.9}$$

Thus we have the sumcheck as

$$\sum_{X \in H} f(X) = \begin{cases} \sum_{k=0 \mod |H|}^{d-1} a_k \cdot |H| & ; d \geq |H| \\ |H| \cdot a_0 = |H| \cdot f(0) & ; \ d < |H| \end{cases} \tag{1.10}$$

which means that to satisfy (1.6) we need to have the relations

$$\sum_{k=0 \mod n}^{d} a_k = \frac{\sigma}{|H|} \ ; \ d \geq |H| \tag{1.11}$$

$$a_0 = \frac{\sigma}{|H|} \ ; \ d < |H| \tag{1.12}$$

or the prover needs to "know" this coefficient, which is the essential verification issue in the sumcheck problem.

For future reference for the case $d < |H|$ ((1.1) we can rewrite (1.11) using (1.4),(1.5) as

$$\sum_{i=0}^{n-1} v_i = \sigma = |H| \cdot a_0 = |H| \cdot f(0) \tag{1.13}$$

1.2 Overview: Proving the sumcheck in monomial basis $d \geq |H|$

In this section, we provide an overview of proving the sumcheck. Note that the RHS of (1.10) is a constant, and in general the summand in the LHS of (1.10) does not vanish. Thus the summand must be proportional to a term that vanishes in the domain and a term that doesn't.

The term that vanishes in a domain H is always proportional to the vanishing polynomial. We recollect that the vanishing polynomial in a domain H is defined as the monic

$$v_H(X) = X^H - 1 = \prod_{k=0}^{n-1} (X - x_k) \ \forall \ x_k \in H \tag{1.14}$$

that vanishes only on all points within H, and is non-vanishing outside. [1] Thus if we divide the summand in (1.6) by the vanishing polynomial it must have a finite remainder, which we express as

$$f(X) = h(X) \cdot v_H(X) + r(X) \tag{1.16}$$

While we could use this equation for the sumcheck, by requiring the prover to send the polynomials $h(X)$ with $deg(h(X)) = d - |H|$ and $r(X)$ with $deg(r(X)) < |H| - 1$, it allows a malicious prover to manipulate the constant coefficients since it hides in $r(X)$. i.e

$$\sum_{X \in H} f(X) = \sum_{X \in H} h(X) \cdot v_H(X) + \sum_{X \in H} r(X) \tag{1.17}$$

(see following section §2). Observing that the $V_H(X)$ vanishes in H, we find that the constant terms come from the Remainder polynomial

$$\sum_{X \in H} r(X) = \sigma \tag{1.18}$$

which means that $r(X)$ must have the structure

$$r(X) = X \cdot g(X) + \frac{\sigma}{|H|} \tag{1.19}$$

where $deg(g) < |H| - 1$. The X is multiplied to $g(X)$ to prevent malleability of the constant term (see §2). Substituting the above equation in (1.18) and using (1.9) we see that the relation (1.18) is satisfied. Thus the most general expression for an $f(X)$ that can satisfy (1.6) is given by the expression

$$f(X) = h(X).v_H(X) + X \cdot g(X) + \frac{\sigma}{|H|} \tag{1.20}$$

Taking the sum $\sum_{X \in H}$ on both sides, we see that it indeed reproduces the equation (1.6). However, note that

$$\sum_{X \in H} f(X) = \sigma \equiv |H| \cdot r(0) \tag{1.21}$$

where $r(X) = X \cdot g(X) + \sigma/|H|$. Thus proving that (1.6) sums to a certain value, is equivalent to proving that the prover knows polynomials $h(X), g(X)$ satisfying the necessary degree bounds, and the verifier can check that the polynomial identity (1.20) is satisfied at any random $\alpha \in \mathbb{F}$. This is the main technical result of the univariate sumcheck protocol.

In the literature, the sumcheck implementation varies depending on whether the commitment scheme and cryptographic assumptions. If the commitment scheme is Merkle, then the sumcheck proceeds with enforcing the degree bounds on $g(X)$ and $h(X)$ via a low degree

[1] Thus any equation $C(X) = 0, X \in H$ can be written as a polynomial identity

$$C(X) = q(X) v_H(X) \tag{1.15}$$

If $C(X)$ is of degree D, then it follows that $q(X)$ can be at most of degree $D - |H|$. We refer to $q(X)$ as the quotient polynomial that results when we divide $f(X)$ by the vanishing polynomial $v_H(X)$.

test. This is for example the case in Aurora [2]. The prover commits to $f(X), g(X), h(X)$ via a Merkle commitment scheme, the verifier can simulate the quotient computation

$$h(X) := \frac{f(X) - X \cdot g(X) - \sigma/|H|}{v_H(X)} \tag{1.22}$$

by querying f, g at a $\alpha \in \mathbb{F}$ and computing the right hand side of (1.22), and comparing the result with h at $\alpha \in \mathbb{F}$. If the prover did not commit the correct $f(X), g(X)$ the vanishing polynomial will not divide the RHS of (1.22) and then in that case the LHS will necessarily be a rational function $h(X) = p(X)/q(X)$.

These facts are checked using FRI as follows: on the LHS is the verifier does a Low Degree Test (LDT) on $h(X)$ and checks if it is of degree $< d - |H|$ and on the RHS the verifier does a LDT on $g(X)$ to check if it satisfies the degree bound $< |H| - 1$. Basically for each folding in the FRI, (1.22) is a linear relation that is satisfied by the path elements in the LDT.

In the case that, the commitment scheme is KZG [3] first the prover commits to the polynomials $f(X), g(X), h(X)$ in a prime order subgroup $G_1 : E/F_q$ which we refer to as $[f]_1, [g]_1, [h]_1$. The verifier then checks the pairing equation at a random $\gamma \leftarrow \mathbb{F}$ by querying the polynomials f, g, h at γ and checks the polynomial identity via a pairing equation

$$e([f(\gamma) - \sigma/|H|]_1, [1]_2) = e([h(\gamma)]_1, [v_H(\gamma)]_2) \cdot e([g(\gamma)]_1, [\gamma]_2) \tag{1.23}$$

However, note that this does not do any low degree testing of g, h. However, the prover must provide KZG opening proofs by computing the quotient polynomials $[T_f(\gamma)]_1, [T_g(\gamma)]_1, [T_h(\gamma)]_1$ for opening of f, g, h at γ. This further entails 3 pairing checks on the verifier end.

To summarize, the univariate sumcheck (1.6) the protocol proceeds as follows in table 1.

Prover	**Communication**	**Computation**
Compute coefficient form $f(X)$	commit to $f(X)$	MSM/Merkle root
Compute $g(X), h(X)$ in (1.20)	commit to $g(X), h(X)$	MSM/Merkle root
Setup	**Commitment scheme**	**Verification Method**
Transparent	Merkle commitment	Low degree testing for g, h
Trusted	KZG commitment	Pairing+KZG opening proofs for g, h

Table 1. Univariate sumcheck for trusted/transparent proof systems

2 Subtle soundness in univariate sumcheck

In this section we motivate a soundness issue in how one implements the univariate sumcheck. It is subtle and can easily escape notice. Let us begin with

$$f(X) = h(X) \cdot V_H(X) + r(X) \tag{2.1}$$

and we want to verify that $\sum_{X \in H} f(Y) = \sigma$, $r(X)$ is the remainder after dividing $f(X)$ with the vanishing polynomial and that $r(0) = \sigma/|H|$. Suppose the protocol proceeds as follows

- The prover computes $h(X)$ and $g(X) = r(X) - r(0)$ and sends them to the verifier.
- The verifier generates a random challenge $\alpha \in \mathbb{F}$ and sends it to the prover.
- The prover evaluates $f(\alpha), g(\alpha), h(\alpha)$ and sends them to the verfier.
- The verifier checks

$$f(\alpha) \stackrel{?}{=} h(\alpha) \cdot V_H(\alpha) + g(\alpha) + \frac{\sigma}{|H|} \tag{2.2}$$

Suppose there exists a Malicious prover whose sum evaluates to $\sum_{X \in H} f(X) = \sigma'$. Since in the check (2.4), g is used as it is sent by the prover, He/she can trick the verifier into passing (2.4) by manipulating the constant term in the function g. For instance, if the Malicious prover constructs

$$\tilde{g}(X) = \frac{\sigma' - \sigma}{|H|} + g(X) \tag{2.3}$$

evaluates it in the random value α sent by the verifier, then the verifier will be checking

$$\begin{aligned} f(\alpha) &\stackrel{?}{=} h(\alpha) \cdot V_H(\alpha) + \tilde{g}(\alpha) + \frac{\sigma}{|H|} \\ &\stackrel{?}{=} h(\alpha) \cdot V_H(\alpha) + g(\alpha) + \frac{\sigma'}{|H|} \end{aligned} \tag{2.4}$$

where we substituted (2.3) in the second line. The check will pass because the $\sum_{X \in H} f(X) = \sigma'$ in the malicious prover computation. Thus in the univariate sumcheck, the soundness issue is resolved by multiplying the $g(X)$ sent by the prover with X as in (1.20). This prevents any manipulation of the constant term in the RHS of (1.20), and the check will truly pass if and only if $\sum_{X \in H} f(X) = \sigma$.

Acknowledgments

We stand on the shoulders of giants.

A Lagrange basis

For this we define the Lagrange basis in a domain H spanned by the set of roots of unity $x_i \equiv \{1, \omega, \omega^2, \dots \omega^{n-1}\}$ as

$$L_{H,x_i}(X) = \frac{\prod_{k=0, k \neq i}^{n-1} (X - x_k)}{\prod_{k=0, k \neq i}^{n-1} (x_i - x_k)} \tag{A.1}$$

that satisfy the relations

$$L_{H,x_i}(X = x_j) = \delta_{ji} = \begin{cases} 1 \ , \ j = i \ \forall \ i, j = 0, 1, \dots n-1 \\ 0 \ , \ \text{otherwise} \end{cases} \tag{A.2}$$

Note that each basis polynomial (A.1) is at most of degree $n-1$. We define a (Low Degree Extension) LDE of any given set of n points v_i in terms of the Lagrange polynomial (A.1)

$$F(X) = \sum_{i=0}^{n-1} v_i \cdot L_{H,x_i}(X) \tag{A.3}$$

which is a polynomial of at most $deg \leq n-1$. This is referred to as the evaluations form of the polynomial. We recollect that the vanishing polynomial in a domain H is defined as the monic

$$v_H(X) = X^H - 1 = \prod_{k=0}^{n-1}(X - x_k) \; \forall \; x_k \in H \tag{A.4}$$

that vanishes only on all points within H, and is non-vanishing outside.

References

[1] E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward, "Aurora: Transparent succinct arguments for r1cs." Cryptology ePrint Archive, Report 2018/828, 2018. https://ia.cr/2018/828.

[2] E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward, "Aurora: Transparent succinct arguments for r1cs." Cryptology ePrint Archive, Paper 2018/828, 2018. https://eprint.iacr.org/2018/828.

[3] A. Kate, G. M. Zaverucha, and I. Goldberg, *Constant-size commitments to polynomials and their applications*, in *Advances in Cryptology - ASIACRYPT 2010* (M. Abe, ed.), (Berlin, Heidelberg), pp. 177–194, Springer Berlin Heidelberg, 2010.