

Lecture 11

Instructor: Sruthi Sekar

Scribe: Pratham Garg

1 Multivariate Polynomials

1.1 Recap

- Fundamental properties and definition of fields, including fields of prime order.
- Multivariate polynomials and their key characteristics.
- Extension fields and their significance in constructing fields of non-prime order.

1.2 Fields

An **field**, is a set F equipped with two binary operations: addition $(+)$ and multiplication $(\cdot)$, satisfying certain axioms. Formally, a field $(F, +, \cdot)$ satisfies the following properties: (1) $(F, +)$ is an abelian group with identity element 0, (2) $(F \setminus \{0\}, \cdot)$ is an abelian group with identity element 1, (3) multiplication distributes over addition, i.e., for all $a, b, c \in F$, we have $a \cdot (b + c) = a \cdot b + a \cdot c$. Common finite fields are prime modulo fields.

1.3 Multivariate Polynomials

A polynomial $p(X_1, \dots, X_i) \in \mathbb{F}[X_1, \dots, X_i]$ computes a function $f : \mathbb{F}^i \rightarrow \mathbb{F}$. Conversely, every function $f : \mathbb{F}^i \rightarrow \mathbb{F}$ is computable using a polynomial.

For any function $f : \mathbb{F}_q^i \rightarrow \mathbb{F}_q$, there exists a unique polynomial representation in which the degree of each monomial is at most $q - 1$. This result is fundamental in the algebra of finite fields.

1.4 Reduced Polynomial Representation

Number of functions from $\mathbb{F}_q^n$ to $\mathbb{F}_q$ is q^{q^n} . But the set of polynomials is infinite, hence multiple polynomials actually represent the same function.

Every monomial is hence reduced as follows :

$$X^m \equiv X^{m \bmod (q-1)}.$$

Hence number of reduced monomials is bounded by q^n , thus ensuring a unique reduced polynomial representation for every function.

2 Schwartz-Zippel Theorem

The **Schwartz-Zippel theorem** is a fundamental result in algebraic complexity and probabilistic algorithms, used to bound the probability that a nonzero polynomial evaluates to zero at a randomly chosen point. It is particularly useful in randomized algorithms for polynomial identity testing.

Theorem 1 (Schwartz-Zippel). *Let $p(X_1, X_2, \dots, X_n)$ be a nonzero polynomial of total degree d over a field $\mathbb{F}$. Suppose we choose values $x_1, x_2, \dots, x_n$ independently and uniformly at random from a finite subset $S \subseteq \mathbb{F}$. Then, the Schwartz-Zippel theorem states:*

$$\Pr[p(x_1, x_2, \dots, x_n) = 0] \leq \frac{d}{|S|}.$$

Proof. We prove the Schwartz-Zippel theorem by induction on the number of variables n . Let $p(X_1, X_2, \dots, X_n)$ be a nonzero polynomial of total degree d over a field $\mathbb{F}$, and let S be a finite subset of $\mathbb{F}$. The theorem states that if $x_1, x_2, \dots, x_n$ are chosen independently and uniformly at random from S , then:

$$\Pr[p(x_1, x_2, \dots, x_n) = 0] \leq \frac{d}{|S|}.$$

Base Case ($n = 1$): When $p(X)$ is a univariate polynomial of degree at most d , it has at most d roots. Since x is selected randomly from S , the probability that $p(x) = 0$ is at most:

$$\Pr[p(x) = 0] \leq \frac{d}{|S|}.$$

Inductive Step: Assume the theorem holds for polynomials with $n - 1$ variables. Consider a polynomial $p(X_1, X_2, \dots, X_n)$ of degree d . We can express p as a univariate polynomial in X_n , where the coefficients are polynomials in $X_1, X_2, \dots, X_{n-1}$:

$$p(X_1, X_2, \dots, X_n) = \sum_{i=0}^{d'} q_i(X_1, X_2, \dots, X_{n-1}) X_n^i,$$

where each q_i is a polynomial in $n - 1$ variables, and $d' \leq d$ is the degree of p in X_n . Since p is nonzero, at least one q_i is a nonzero polynomial with degree at most $d - i$.

Now, we first choose $x_1, x_2, \dots, x_{n-1}$ randomly from S . By the induction hypothesis, the probability that all $q_i(x_1, x_2, \dots, x_{n-1}) = 0$ is at most $\frac{d-i}{|S|}$. If at least one q_i is nonzero, then $p(x_1, x_2, \dots, X_n)$ becomes a univariate polynomial in X_n of degree at most d' , and by the base case:

$$\Pr[p(x_1, x_2, \dots, x_n) = 0 \mid x_1, x_2, \dots, x_{n-1}] \leq \frac{d'}{|S|}.$$

Taking the expectation over all choices of $x_1, x_2, \dots, x_{n-1}$, we obtain:

$$\Pr[p(x_1, x_2, \dots, x_n) = 0] \leq \frac{d}{|S|}.$$

Thus, by induction, the theorem holds for all n . This completes the proof. $\square$

3 Error Correction

Error correction codes are techniques used to detect and correct errors in data transmission or storage. These codes add redundancy to the original data, allowing errors to be identified and corrected without needing retransmission.

3.1 Rate of an Error Correction Code

The **rate** of an error correction code measures the efficiency of the code in terms of the amount of redundant information added. It is defined as the ratio of the number of information symbols to the total number of symbols in a codeword. Formally, if a code encodes k information symbols into n symbols (where $n > k$), the rate R of the code is given by:

$$R = \frac{k}{n}.$$

A higher rate ($R \approx 1$) means less redundancy and higher information efficiency, while a lower rate ($R \ll 1$) indicates more redundancy, providing greater error correction capability. The choice of code rate depends on the application, balancing error resilience and data transmission efficiency.

3.2 Hamming Distance and Hamming Ball

The **Hamming distance** between two strings of equal length is the number of positions at which their corresponding symbols differ. Formally, for two strings $x, y \in \mathbb{F}_q^n$, the Hamming distance $d_H(x, y)$ is defined as:

$$d_H(x, y) = |\{i \mid x_i \neq y_i, 1 \leq i \leq n\}|.$$

For example, consider the binary strings $x = 1011101$ and $y = 1001001$. The Hamming distance between them is $d_H(x, y) = 2$ since they differ in the third and fifth positions.

A **Hamming ball** of radius r centered at a point x in $\mathbb{F}_q^n$ consists of all points whose Hamming distance from x is at most r . It is defined as:

$$B_H(x, r) = \{y \in \mathbb{F}_q^n \mid d_H(x, y) \leq r\}.$$

For example, in the binary space $\mathbb{F}_2^3$, the Hamming ball of radius $r = 1$ centered at $x = 000$ includes the points $\{000, 001, 010, 100\}$. Hamming balls are crucial in error correction, as they represent the set of received words that can be decoded to a given codeword.

3.3 Minimum Hamming Distance

The **minimum Hamming distance** of an error-correcting code is the smallest Hamming distance between any two distinct codewords in the code. It determines the error-detecting and error-correcting capability of the code. If a code $\mathcal{C}$ has a minimum Hamming distance $d_{\min}$, then it can detect up to $d_{\min} - 1$ errors and can correct up to $\lfloor (d_{\min} - 1)/2 \rfloor$ errors.

3.4 Example

For example, consider a code with codewords $\{000, 111\}$. The minimum Hamming distance between these codewords is 3. This means the code can detect up to 2 errors and correct up to $\lfloor 2/2 \rfloor = 1$ error. In error correction, if a received word differs from a valid codeword by at most $\lfloor (d_{\min} - 1)/2 \rfloor$, it is decoded as the nearest codeword in terms of Hamming distance.

3.5 Linear Codes, Generator Matrix, and Code Space

A **linear code** is a type of error-correcting code where the set of codewords forms a vector space over a finite field $\mathbb{F}_q$. An (n, k) linear code is a subspace of $\mathbb{F}_q^n$ with dimension k , meaning it consists of q^k codewords. The main advantage of linear codes is that they allow efficient encoding and decoding operations.

A linear code can be described using a **generator matrix** G , which is a $k \times n$ matrix over $\mathbb{F}_q$. Each row of G forms a basis for the code space, and any message $m \in \mathbb{F}_q^k$ is encoded into a codeword c using matrix multiplication:

$$c = mG.$$

The set of all possible codewords generated by G forms the **code space**, which is a subspace of $\mathbb{F}_q^n$. The structure of the code space ensures that linear combinations of codewords remain valid codewords, simplifying both error detection and correction. Linear codes, such as Hamming codes and Reed-Solomon codes, are widely used in digital communication and data storage for their efficient error-correcting capabilities.

3.6 Dual Code and Parity-Check Matrix

Given an (n, k) linear code $\mathcal{C}$ over a finite field $\mathbb{F}_q$, the **dual code** $\mathcal{C}^\perp$ is the set of all vectors in $\mathbb{F}_q^n$ that are orthogonal to every codeword in $\mathcal{C}$ under the standard dot product. Formally, the dual code is defined as:

$$\mathcal{C}^\perp = \{v \in \mathbb{F}_q^n \mid v \cdot c = 0 \text{ for all } c \in \mathcal{C}\}.$$

The dual code $\mathcal{C}^\perp$ is an $(n, n - k)$ linear code and plays a crucial role in error detection and syndrome decoding.

A key tool for describing the dual code is the **parity-check matrix** H , which is an $(n - k) \times n$ matrix whose rows form a basis for $\mathcal{C}^\perp$. The parity-check matrix satisfies the fundamental property:

$$Hc^T = 0, \quad \forall c \in \mathcal{C}.$$

This property is used in error detection: if a received word r is a valid codeword, then $Hr^T = 0$.

4 Conclusion

This lecture delved into key concepts related to multivariate polynomials, emphasizing their significance in polynomial identity testing and their practical implications in error-correcting codes. Additionally, the discussion covered the foundational principles of the Schwartz-Zippel theorem, highlighting its role in probabilistic algorithms and its impact on computational complexity.