CSCI-B609: A Theorist's Toolkit, Fall 2016 Oct 4

Lecture 14: Schwartz-Zippel Lemma and Intro to Coding Theory

Lecturer: Yuan Zhou *Scribe: Haoyu Zhang*

1 Roots of polynomials

Theorem 1. *A non-zero, univariate polynomial with degree d has at most d roots.*

Proof. Given a polynomial $P(x)$, we could write it as:

$$P(x) = (x-\alpha_1)(x-\alpha_2)(x-\alpha_3)\dots(x-\alpha_k)(x^2+\dots)(x^3+\dots)\dots$$

where $(x^2+\dots)$,$(x^3+\dots)$ are irreducible and α_i are roots. Note that irreducible polynomial has no root, because if an irreducible polynomial $A(x)$ has a root β, we can write $A(x) = (x-\beta)Q(x) + R$. β is a root if and only if $R \equiv 0$, which implies that $(x-\beta)|Q(x)$. □

2 Computation complexity

Now we present an application of the Theorem 1.

Alice	*Bob*
$a_1, a_2 \dots a_n \in \{0,1\}^n$	$b_1, b_2 \dots b_n \in \{0,1\}^n$

The description of the problem is: Alice and Bob each holds an n-bits vector $a_{1\dots n}$, $b_{1\dots n}$, and they want to know whether the vectors are same. Alice and Bob could send messages to each other, the goal is to minimize the size of communication between Alice and Bob. A trivial protocol is: Alice sends the whole n-bits to Bob, then Bob checks whether the two vectors are same and sends the result (Yes or No) to Alice. The whole communication uses $n+1$-bits message.

In fact, for any deterministic protocol which answers that question exactly, it needs at least $\Omega(n)$ bits of communication. However, we could find a randomized protocol with smaller communication cost. The new protocol is as following:

[1] Alice fixes the $\mathbb{F}_p$ by choosing a prime $p \in [n^2, 2n^2]$ and sends it to Bob.

[2] Alice(Bob) forms the $A(x) = a_1 + a_2x \dots a_nx^{n-1}$ ($B(x) = b_1 + b_2x \dots b_nx^{n-1}$).
[3] Alice chooses uniformly $\alpha \in \mathbb{F}_p$, and sends α and $A(\alpha)$ to Bob.
[4] Bob calculates $B(\alpha)$ and reports whether $A(\alpha) = B(\alpha)$(Yes or No) to Alice.

The protocol uses $O(\log(n))$ bits and succeeds with probability at least $1 - \frac{1}{n}$. The reason is, the protocol fails only when $a_{1\dots n} \neq b_{1\dots n}$ and $A(\alpha) = B(\alpha)$. Consider the following polynomial $C(\alpha) = A(\alpha) - B(\alpha)$ which has at most $n-1$ roots. Then the probability of the protocol fails is:

$$\Pr[\text{The protocol fails}] \leq \Pr[C(\alpha) = 0] \leq \frac{n-1}{p} < \frac{1}{n}$$

3 Schwartz-Zippel Lemma

Lemma 1. *Let $P \in \mathbb{F}_q[x_1 \dots x_n]$ be a non-zero polynomial with total degree at most d, then:*

$$\Pr_{\alpha_1,\dots,\alpha_n \sim \mathbb{F}_q}[P(\alpha_1, \dots, \alpha_n) \neq 0] \geq \frac{1}{q^{\lfloor \frac{d}{q-1} \rfloor}} \left(1 - \frac{d \mod (q-1)}{q}\right)$$

The general description of the lemma is quite complicated. Let us look at two useful special cases –

Case 1: $q = 2$. The right hand side simplifies to be 2^{-d}.

Case 2: $d < q-1$. The right hand side becomes $1 - \frac{d}{q}$, then the whole inequality becomes:

$$\Pr[P(\alpha_1, \dots, \alpha_n) = 0] \leq \frac{d}{q}.$$

If we choose α's from a subset of $S \subset \mathbb{F}$, we have the more commonly stated version:

$$\Pr[P(\alpha_1, \dots, \alpha_n) = 0] \leq \frac{d}{|S|}.$$

The proof of Schwartz-Zippel Lemma is given as our assignment (It could be proof by induction). For the base case $n = 1$, it is just the Theorem 1.

4 Perfect matching in bipartite graphs

We now present an application of Lemma 1.

Goal: Given a bipartite graph $G = (U, V, E)$ check if there is a perfect matching.

Given G, we first create the *Tuttle matrix* [Tut47] in the following way:

$$A(x_{ij}) = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}, \text{where } a_{ij} = \begin{cases} x_{ij} & \text{if}(i,j) \in E \\ 0 & \text{otherwise} \end{cases}.$$

Note that x_{ij} are variables. Let $P(x_{ij}) = \det(A(x_{ij}))$ be a polynomial of degree at most n.

Claim 1. *$P \equiv 0$ if and only if G does not have a perfect matching.*

Proof. If G has a perfect matching M. Let $x_{ij} = \mathbb{1}[(i,j) \in M]$, we have $P(x_{ij}) = 1$. If G has no perfect matching,

$$P(x_{ij}) = \sum_{\text{all permutation } \sigma:[n]\to[n]} (-1)^{C_\sigma} \Pi_{i=1}^{n} a_i \sigma_{(i)} = 0.$$

C_σ is a function of σ which we do not care about it. Note that $P \in \mathbb{Q}[x_1 \dots x_n]$. Let us choose each $x_{ij} \in \{1, 2, \dots, n^2\}$ uniformly:

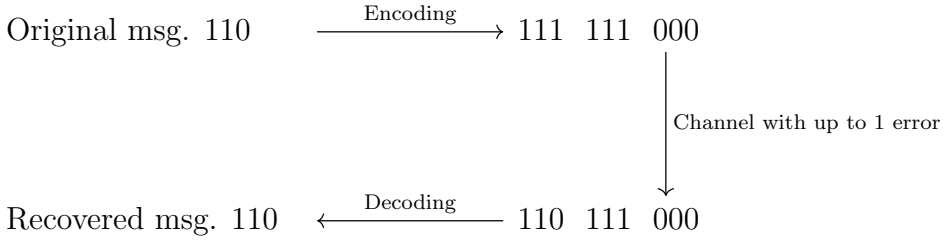
If $P \equiv 0$, we have $\Pr[P(x_{ij}) = 0] = 1$.

If $P \equiv 1$, we have $\Pr[P(x_{ij}) = 0] \leq \frac{\deg(P)}{|S|} = \frac{n}{n^2} = \frac{1}{n}$. □

Lastly, the problem of computing the determinant over $\mathbb{Q}$ is in *NC* [Csa76].

5 Basic definition of Error-correcting codes

Background and Setup: Many communication channels are subject to channel noise, and thus we want to use some redundancy codes to correct errors. The applications are including but not limited to DVDs and sending message to space probes. Here is one simple example (repetition codes):



Definition 1. *the general setup of error correcting codes are:*

$$x \in \Sigma^k \xrightarrow{Enc} x \in \Sigma^n \xrightarrow[Upto\ t\ errors]{Noisy\ channel} x \in \Sigma^n \xrightarrow{Dec} x \in \Sigma^k$$

where:

- *Enc: the encoding function, which maps* $x \in \Sigma^k \to x \in \Sigma^n$*, and it has to be efficient*
- *Dec: the decoding function,which maps* $x \in \Sigma^n \to x \in \Sigma^k$*, and it has to be efficient*
- Σ*: alphabet of size* q
- Σ^k*: is the message space*
- k*: message dimension*
- n*: block length*
- $C = Im(Enc)$*: code*
- $y \in C$*: codeword*
- $\frac{k}{n}$*: rate of code(want it high)*

Then we want to ask: how to measure the ability to tolerate errors? Before answering that, let's come to a few more definitions.

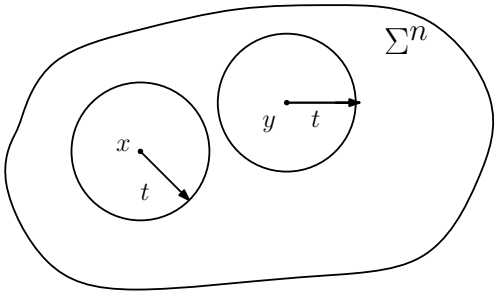


Figure 1: Hamming Ball

Definition 2. *Let* $x, y \in \Sigma^k$*, the* **hamming distance** *of* x, y *is*

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

The **minimum distance** *of a code* C *is* $d = \min_{x \neq y\ x,y \in C} \triangle(x, y)$.

The **relative distance** *of a code* C *is* $\frac{d}{n}$.

Fact 1. *Suppose the channel introduce at most t errors. Then we can uniquely decode if and only if $t \leq \lfloor \frac{d}{2} \rfloor$.*

Proof. Considering the Hamming ball of radius t centered at each codeword. we can uniquely decode the message if and only if the Hamming balls do no intersect. □

References

[Csa76] Csanky, L. Fast parallel matrix inversion algorithms. *SIAM Journal on Computing*, 1976, 5(4): 618–623.

[Tut47] Tutte, W T. The factorization of linear graphs. *Journal of the London Mathematical Society*, 1947, 1(2): 107–111.