6.842 Randomness and Computation February 18, 2022

Lectures on Polynomial Identity Testing

Lecturer: Ronitt Rubinfield *Scribe: Yuchong Pan*

1 Univariate Polynomial Identity Testing

Polynomial identity testing asks whether two polynomials are identical, e.g., $P(x) = (x+3)^{38}(x-4)^{83}$ and $Q(x) = (x-4)^{38}(x+3)^{83}$? The following two problems are equivalent:

Problem 1. *Given two polynomials P, Q of degree at most d, is $P \equiv Q$?*

Problem 2. *Given two polynomials P of degree at most d, is $P \equiv 0$?*

Recall the following fundamental fact of algebra:

Fact 3. *If R is a polynomial of degree at most d with $R \not\equiv 0$, then R has at most d roots.*

Fact 3 implies a simple algorithm to test if a polynomial is identical to 0, given in Algorithm 1. This algorithm requires $O(d)$ evaluations of R.

pick $d+1$ points $x_1, \ldots, x_{d+1}$
if $R(x_i) = 0\ \forall i \in [d+1]$ **then**
    output "$R \equiv 0$"
else
    // $\exists i \in [d+1]$ s.t. $R(x_i) \neq 0$
    output "$R \not\equiv 0$"

Algorithm 1: A simple algorithm for testing whether a polynomial R of degree at most d is identical to 0.

Indeed, we can design a faster randomized algorithm to test if a polynomial is identical to 0, given in Algorithm 2. If $R \equiv 0$, the algorithm will always output "$R \equiv 0$." If $R \not\equiv 0$, in each loop,

$$\mathbb{P}_{i\in[2d]}\left[R\left(x_i\right) = 0\right] \leq \frac{\#\text{ roots}}{2d} \leq \frac{d}{2d} = \frac{1}{2}.$$

Therefore,

$$\mathbb{P}[\text{error}] = \mathbb{P}[\text{choose roots in all } k \text{ iterations}] \leq \frac{1}{2^k}.$$

It follows that

$$\mathbb{P}[\text{output "}R \not\equiv 0\text{"}] \geq 1 - \frac{1}{2^k}.$$

If we tolerate the probability of error to be at most δ, then we pick $k = \log 1/\delta$ (which does not depend on d), and the probability of outputting "$R \not\equiv 0$" is at least $1 - \delta$.

```
pick 2d distinct points x_1, ..., x_{2d}
repeat k times
    pick i ∈ [2d]
    if R(x_i) ≠ 0 then
        output "R ≠ 0"
        return
output "R ≡ 0"
```

Algorithm 2: A faster randomized algorithm for testing whether a polynomial R if degree at most d is identical to 0.

1.1 Application: Person on the Moon

The "person on the moon" problem is formulated as follows: A person on the earth has an $(n+1)$-bit string $w = w_0 \dots w_n$. A person on the moon has another $(n+1)$-bit string $w^* = w_0^* \dots w_n^*$. The question is whether $w = w^*$.

Let

$$P(x) = w_n x^n + w_{n-1} x^{n-1} + \dots + w_1 x + w_0,$$
$$P^*(x) = w_n^* x^n + w_{n-1}^* x^{n-1} + \dots + w_1^* x + w_0^*.$$

Then P and P^* are polynomials of degree n. Moreover, $w = w^*$ if and only if $P \equiv P^*$. Here is the general strategy motivated by polynomial identity testing:

- The earth person picks random $r_1, \dots, r_k$ and sends

$$(r_1, P(r_1)), \qquad (r_2, P(r_2)), \qquad \dots, \qquad (r_k, P(r_k)).$$

- The moon person checks equality on $r_1, \dots, r_k$.

Each r_i for $i \in [k]$ requires $\Theta(\log n)$ bits of communication. However, a polynomial of degree n evaluated on an input of $\Theta(\log n)$ bits can give values as large as $\Theta(n^n)$ (which rquires $\Theta(n \log n)$ bits to describe). To fix this issue, we can use a prime q of $O(\log n)$ bits and send every number modulo q. Therefore, the total number of bits is $O(k \log n)$.

2 Multivariate Polynomial Identity Testing

In this section, we study multivariate polynomial identity testing:

Problem 4. *Given a multivariate polynomial R on n variables $x_1, \dots, x_n$, is $R \equiv 0$?*

Definition 5. Given a multivariate polynomial R on n variables $x_1, \dots, x_n$, the *total degree* of R is defined to be

$$\max_{s\text{: term of } R} (\text{sum of degrees of the } x_i\text{'s in } s).$$

Multivariate polynomial identity testing have the following two difficulties:

- *Difficulty 1:* A multivariate polynomial $R \not\equiv 0$ can have infinitely many roots, e.g., $R(x, y) = x \cdot y$ and $R(x, y) = x - y$.
- *Difficulty 2:* The number of terms in a multivariate polynomial of total degree d can be $\binom{n}{d}$.

Theorem 6 (Schwartz-Zippel-De Mill Lipton)**.** *Let R be a multivariate polynomial of degree d on n variables $x_1, \ldots, x_n$ such that $R \not\equiv 0$. Let S be a set of points such that $|S| = 2d$. Pick $x_i \in_R$ for all $i \in [n]$. Then*

$$\mathbb{P}\left[R\left(x_1, \ldots, x_n\right) = 0\right] \leq \frac{d}{|S|}.$$

2.1 Application: Bipartite Perfect Matching

Definition 7. Let $G = (A \sqcup B, E)$ be a bipartite graph. A *matching* is a subset $M \subseteq E$ in which no two edges share a vertex. A *perfect matching* is a matching M such that $|V[M]| = n$.

Definition 8. Let $G = (A \sqcup B, E)$ be a bipartite graph. Let $x_{u,v}$ be a variable for each $u \in A, v \in B$. The *Tutte matrix* of G is defined to be an $|A| \times |B|$ symbolic matrix $A = (a_{u,v})_{u \in A, v \in B}$ such that

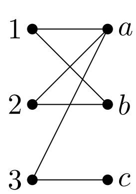
$$a_{u,v} = \begin{cases} x_{u,v}, & \text{if } (u,v) \in E, \\ 0, & \text{otherwise.} \end{cases}$$

Example. Let G be the graph in Figure 1a. Figures 1b and 1c give two perfect matchings in G. Moreover, the Tutte matrix A_G of G is given by

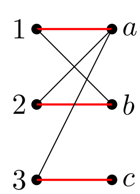
$$A_G = \begin{pmatrix} x_{1,a} & x_{1,b} & 0 \\ x_{2,a} & x_{2,b} & 0 \\ x_{3,a} & 0 & x_{3,c} \end{pmatrix},$$

where the rows are indexed by the vertices on the left side, and the columns are indexed by the vertices on the right side. We have

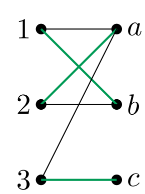
$$\det A_G = x_{1,a}x_{2,b}x_{3,c} - x_{1,b}x_{2,a}x_{3,c}.$$



(a) The graph G.



(b) A perfect matching in G.



(c) Another perfect matching in G.

Figure 1: An example of matchings and the Tutte matrix.

Proposition 9. *A bipartite graph G has a perfect matching if and only if* $\det A_G \not\equiv 0$.

Proof. Recall that

$$\det A_G = \sum_{\sigma\text{: permutation of } n} \operatorname{sign}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}.$$

Note that $\operatorname{sign}(\sigma)$ is either 1 or -1, and that $a_{i,\sigma(i)}$ is either $x_{i,\sigma(i)}$ or 0. The main insight is that a permutation of $[n]$ corresponds to a potential perfect matching such that $(i, \sigma(i))$ is a potential edge in the matching. For each permutation σ of $[n]$, $\prod_{i=1}^{n} a_{i,\sigma(i)} = 0$ if one edge in the corresponding potential matching is missing, so $\prod_{i=1}^{n} a_{i,\sigma(i)} \neq 0$ if and only if it corresponds to a perfect matching. Therefore, $\det A_G \not\equiv 0$ if and only if there exists one permutation σ of $[n]$ which corresponds to a perfect matching. $\square$

Note that $\det A_G$ is a polynomial on n^2 variables (one for each possible edge) of total degree n. Therefore, Proposition 9 gives Algorithm for testing whether a bipartite graph has a perfect matching. The total number of terms in $\det A_G$ is at most $n!$. However, we can use matrix multiplication algorithms to compute the determinant of a matrix in time that is polynomial in n.

```
A_G ← the Tutte matrix of G
test whether det A_G ≢ 0
if det A_G ≢ 0 then
    output "G has a perfect matching"
else
    output "G does not have a perfect matching"
```

Algorithm 3: An algorithm for testing whether a bipartite graph G has a perfect matching.