Mathematical Toolkit **Autumn 2021**

Lecture 12: November 9, 2021

Lecturer: Madhur Tulsiani

1 Randomized polynomial identity testing

We use our knowledge of events and conditioning, to prove the following lemma, which gives an algorithm for testing if a polynomial f in n variables $x_1, \ldots, x_n$ over a field $\mathbb{F}$ is identically zero. While this is usually referred to as the Schwartz-Zippel lemma, or the DeMillo-Lipton- Schwartz-Zippel lemma, it actually has a longer history as described in (Section 3.1 of) this article by Arvind et al. [AJMR19]. We refer to it as the polynomial identity lemma.

Lemma 1.1 (Polynomial identity lemma) *Let $f(x_1, x_2, \ldots, x_n)$ be a non-zero polynomial of degree $d \geq 0$, i.e.,*

$$\begin{aligned} f(x_1, x_2, \ldots, x_n) &= \sum c_{i_1 i_2 \ldots i_n} \cdot x_1^{i_1} \cdot x_2^{i_2} \cdots x_n^{i_n} \\ s.t., \quad & \quad i_1 + i_2 + \ldots + i_n \leq d \end{aligned}$$

over a field, $\mathbb{F}$. Let $S \subseteq \mathbb{F}$, be a finite subset and let $x_1, x_2, \ldots, x_n$ be selected uniformly at random from S, independently. Then,

$$\mathbb{P}\left[f(x_1, x_2, \ldots, x_n) = 0\right] \leq \frac{d}{|S|}.$$

Proof: We will prove this lemma by induction on n. This lemma can be proved simply by using conditional probability.

Base Case: n = 1

A non zero polynomial, $f(x_1)$ can have at most d roots. Hence, $\mathbb{P}\left[f(x_1) = 0\right] \leq \frac{d}{|S|}$.

Induction Step

Assume that the lemma holds for any polynomial in $n-1$ variables. We need to prove that it holds true for $f(x_1, x_2, \ldots, x_n)$. We can write f as:

$$f(x_1, x_2, \ldots, x_n) = x_1^k \cdot g(x_2, \ldots, x_n) + h(x_1, x_2, \ldots, x_n)$$

where, k is largest degree of x_1. Thus we have $0 < k \leq d$ (if $k = 0$ then we are already done). We also have that $\deg(g(x_2, \ldots, x_n)) \leq d - k$.

Now let us define two events.

$$E \;\equiv\; \{f(x_1, x_2, \ldots, x_n) = 0\} \quad \text{and} \quad F \;\equiv\; \{g(x_2, \ldots, x_n) = 0\}$$

We can then write,

$$\mathbb{P}[E] \;=\; \mathbb{P}[F] \cdot \mathbb{P}[E|F] + \mathbb{P}[\neg F] \cdot \mathbb{P}[E|\neg F]\;.$$

We now analyze each of the terms. By the induction hypothesis, we have

$$\mathbb{P}[F] \;=\; \mathbb{P}[g(x_2, \ldots, x_n) = 0] \;=\; \frac{d-k}{|S|}\,.$$

Also, fixing the values of $x_2 = a_2, \ldots, x_n = a_n$ such that $g(a_2, \ldots, a_n) \neq 0$, $f(x_1, a_2, \ldots, a_n)$ is a degree-k polynomial in x_1. Thus, using the base case, we get that

$$\mathbb{P}[E|\neg F] \;\leq\; \frac{k}{|S|}\,.$$

Bounding the other two probabilities by 1, we get that

$$\mathbb{P}[E] \;\leq\; \frac{d-k}{|S|} \cdot 1 + 1 \cdot \frac{k}{|S|} \;=\; \frac{d}{|S|}$$

as desired. ■

1.1 An application: bipartite perfect matching

Consider the following example which applied the Schwartz-Zippel lemma for testing if a given bipartite graph has a perfect matching. Given a bipartite graph, $G = (U, V, E)$ with $|U| = |V| = n$, we say that the graph has a perfect matching, if there exists a set $E' \subseteq E$ of n edges, with exactly one edge in E' being incident on every vertex of G.

Let us define the Tutte matrix A as

$$A_{ij} = \begin{cases} x_{ij} & \text{if (i, j)} \in E \\ 0 & else \end{cases}$$

Note that A is not necessarily symmetric. The determinant of A can be written as,

$$\mathrm{Det}(A) = \sum_{\pi:[n]\to[n]} \mathrm{sign}(\pi) \prod_{i=1}^{n} A_{i,\pi(i)}$$

where π defines the permutation from rows to columns. Note that the determinant is a degree-n polynomial in the variables x_{ij}. Verify the follwing:

Exercise 1.2 *G has a perfect matching if and only if $Det(A) \not\equiv 0$.*

In this case, computing the determinant is expensive with $n!$ terms. But if we are given the values of the variables x_{ij}, we can simply compute the determinant using the Gaussian elimination method. The degree of the polynomial above is n. Thus, if we assign all variables randomly from a set of $2n$ real values, if $\mathrm{Det}(A) \not\equiv 0$, we will detect it with probability at least $1/2$.

The randomized algorithm given by the polynomial identity lemma can be used to parallelize the checking as well. There is no known deterministic algorithm for this problem which can be parallelized efficiently.

2 The probabilistic method

We now come to very powerful method for proving the existence of several interesting combinatorial objects. The general framework, known as the "probabilistic method" has many variants explored in the beautiful (and highly recommended!) book on the subject by Alon and Spencer [AS08].

We will explore vanilla version of the method, known as the first moment method, which only requires computing expectations. At the heart of it is the simple idea captured by the following proposition.

Proposition 2.1 *Let $X : \Omega \to \mathbb{R}$ be a random variable such that $\mathbb{E}[X] \geq c$ for some $c \in \mathbb{R}$. Then, there exists $\omega \in \Omega$ (with probability measure $\nu(\omega) > 0$) such that $X(\omega) \geq c$.*

Proof: Suppose that for all $\omega \in \Omega$ with $\nu(\omega) > 0$, we have $X(\omega) < c$. Then,

$$\mathbb{E}[X] \;=\; \sum_{\omega\in\Omega} \nu(\omega)\cdot X(\omega) \;<\; \sum_{\omega\in\Omega} \nu(\omega)\cdot c \;=\; c\,,$$

which contradicts the fact that $\mathbb{E}[X] \geq c$. ■

Exercise 2.2 *Prove that if $\mathbb{E}[X] \leq c$, then there exists $\omega \in \Omega$ (with $\nu(\omega) > 0$) such that $X(\omega) \leq c$.*

Exercise 2.3 *Is it true that if $\mathbb{E}[X] = c$, then there exists $\omega \in \Omega$ with $X(\omega) = c$?*

The above simple proposition can yield very interesting results, when the random variable X is set-up properly. In particular, when we want X to measure some property of a combinatorial object, and we set up the distribution such that $\mathbb{E}[X]$ is close to some bound we are interested in, we get that there exists a combinatorial object achieving those bounds. We will see a few examples of this principle.

2.1 A randomized algorithm for Max 3-SAT

Recall that a 3-SAT formula φ is of the form

$$\varphi \equiv C_1 \wedge \cdots \wedge C_m ,$$

where each C_i is a clause of the form $C_i = (l_{i_1} \vee l_{i_2} \vee l_{i_3})$ and each l_{i_j} is in turn x_{i_j} or its negation $\overline{x}_{i_j}$. We assume that each clause contains three *distinct* variables.

In the problem Max 3-SAT, the goal is not necessarily to satisfy all the clauses, but rather find an assignment to the variables which satisfies as many clauses as possible. We show that for any formula φ with m clauses, there exists an assignment satisfying $7m/8$ clauses. Moreover, this can be turned into an algorithm, and one can efficiently find an assignment satisfying $7m/8$ clauses.

Consider assigning each of the variables $x_1, \ldots, x_n$ a value in $\{0,1\}$ independently at random. Let Z be a random variable equal to the number of clauses satisfied by the random assignment. We can write

$$Z = Y_1 + \cdots + Y_m ,$$

where Y_i if the clause C_i is satisfied and 0 otherwise. By linearity of expectation $\mathbb{E}[Z] = \sum_{i=1}^{m} \mathbb{E}[Y_i]$. Note $C_i = (l_{i_1} \vee l_{i_2} \vee l_{i_3})$ is not satisfied if and only if $l_{i_1} = l_{i_2} = l_{i_3} = 0$ which happens with probability $1/8$ since the three literals correspond to three distict variables, which are assigned values 0 and 1 independently with probability $1/2$ each. Thus, $\mathbb{P}[Y_i = 0] = 1/8$, which gives

$$\mathbb{E}[Z] = \sum_{i=1}^{m} \mathbb{E}[Y_i] = \sum_{i=1}^{m} \left(1 - \frac{1}{8}\right) = \frac{7m}{8} .$$

Thus, there *exists* an assignment which satisfies at least $7m/8$ clauses. We now argue that it can be found efficiently. Note that

$$\mathbb{E}[Z] = \frac{1}{2} \cdot \mathbb{E}[Z \mid x_1 = 0] + \frac{1}{2} \cdot \mathbb{E}[Z \mid x_1 = 1] .$$

Thus, at least one of the expectations on the right hand side must be at least $7m/8$. We now need the fact that each of these expectations can be computed efficiently.

Exercise 2.4 *Given access to the 3-SAT formula φ, the expectations $\mathbb{E}[Z \mid x_1 = 0]$ and $\mathbb{E}[Z \mid x_1 = 1]$ can both be computed in time $O(m)$ where m is the number of clauses. Actually, it is also possible to do this in time $O(t)$ if x_1 appears in only t clauses and we are given the list of these clauses.*

Using the above, we can find a value $b_1 \in \{0,1\}$ such that

$$\mathbb{E}[Z \mid x_1 = b_1] \geq \frac{7m}{8} .$$

Continuing similarly by induction, we can find $b_1, \ldots, b_n$ such that

$$\mathbb{E}\left[Z \mid x_1 = b_1, \ldots, x_n = b_n\right] \;\geq\; \frac{7m}{8}\,.$$

Since Z is fixed given the values of all the variables, we get that the assignment $(b_1, \ldots, b_n)$ satisfies at least $7m/8$ clauses.

References

[AJMR19] Vikraman Arvind, Pushkar S. Joglekar, Partha Mukhopadhyay, and S. Raja, *Randomized polynomial-time identity testing for noncommutative circuits*, Theory of Computing **15** (2019), no. 7, 1–36. 1

[AS08] Noga Alon and Joel Spencer, *The probabilistic method*, Wiley-Interscience Series, 2008. 3