Lecture 14: The Schwartz-Zippel Lemma and the Determinant

Anup Rao

May 13, 2021

Randomness vs non-determinism

ALTHOUGH WE cannot show that **BPP** $\subseteq$ **NP**, we can show that an algorithm for $3SAT$ would give a way to simulate all randomized algorithms deterministically. This is captured by the following result:

Theorem 1. $\mathbf{BPP} \subseteq \mathbf{NP}^{\mathsf{SAT}}$.

Proof Suppose $f \in$ **BPP**. Let us first reduce the error of the probabilistic algorithm for f to 2^{-n}. Suppose the algorithm uses m random bits. Thus, we just need to be able to distinguish the case when $M(x,r)$ accepts $1 - 2^{-n}$ fraction of all m bit strings from the case when it accepts only 2^{-n} fraction of all m bit strings. Distinguishing the fractions 1 from 0 would be easy (just try a single string). Distinguishing the fractions 1 from < 1 can be done with a query to SAT. So we shall reduce to this case.

Let $u_1, \ldots, u_k \in \{0,1\}^m$ be k random m bit strings, where k will be chosen to be much smaller than 2^n. Then we have the following claims, where here $r \oplus u_i$ denotes the bitwise parity of the m-bit string r with the m-bit string u_i.

Claim 2. *If $f(x) = 0$, for every choice of $u_1, \ldots, u_k$, there exists some $r \in \{0,1\}^m$ such that $\bigvee_i M(x, r \oplus u_i) \neq 1$.*

The claim following from the union bound. For every choice of $u_1, \ldots, u_k$, if you pick a random r, the probability that $M(x, r \oplus u_i)$ is incorrect is at most 2^{-n}. Thus the probability that any of them is wrong is at most $k2^{-n} < 1$.

In the other case, we prove that the opposite happens:

Claim 3. *If $f(x) = 1$, there exist choices $u_1, \ldots, u_k$, such that for every $r \in \{0,1\}^m$, $\bigvee_i M(x, r \oplus u_i) = 1$.*

For any fixed r, the probability that all choices of u_i fail to give the correct answer is at most 2^{-nk}. Thus, as long as $nk > m$, by the union bound some choice of u_i will work for all choices of r.

Our final algorithm in $\mathbf{NP}^{\mathsf{SAT}}$ is as follows. We start by guessing $u_1, \ldots, u_k$ (say $k = m^2$)to satisfy Claim 3. Then we use the SAT oracle to check whether or not there is an r that makes $M(x, r \oplus u_i)$ accept for some i.

■

Schwartz-Zippel Lemma

Recall that a polynomial $p(x, y, z)$ is an expression of the form

$$14x^2y^5z^8 - 3x^3 + 17y^6z^3.$$

The degree of the polynomial is the maximum of the sums of the powers of the variables in any monomial. So in the last example, the degree is 15.

The Schwartz-Zippel Lemma turns out to be quite useful for randomized algorithms:

Lemma 4. *Let $p(x_1, \ldots, x_n)$ be a polynomial of degree d, such that p is not the 0 polynomial. Let S be any set of numbers, and let $a_1, \ldots, a_n$ be n random numbers drawn from S. Then $\Pr[p(a_1, \ldots, a_n) = 0] \leq d/|S|$.*

Proof We prove the lemma by induction on n. When $n = 1$, the theorem follows from the fact that any non-zero degree d polynomial in one variable has at most d roots. Thus $p(a) = 0$ only when a is a root, which happens with probability at most d.

For the general case. Let us write the polynomial in the form

$$p(x_1, \ldots, x_n) = x_n^\ell \cdot q(x_1, \ldots, x_{n-1}) + r(x_1, \ldots, x_n),$$

where here r is a polynomial in which the degree of x_n is at most $\ell - 1$. So we simply gather all the terms which have maximum degree in x_n.

Now let E_1 be the event that $p(a_1, \ldots, a_n) = 0$, and let E_2 be the event that $q(a_1, \ldots, a_{n-1}) = 0$. Then we have that

$$\begin{aligned}
\Pr[E_1] &= \Pr[E_1 \wedge E_2] + \Pr[E_1 \wedge \neg E_2] \\
&= \Pr[E_2] \cdot \Pr[E_2 | E_1] + \Pr[\neg E_2] \cdot \Pr[E_1 | \neg E_2] \\
&\leq \Pr[E_2] + \Pr[E_1 | \neg E_2].
\end{aligned}$$

By induction, since q is a degree $d - \ell$ polynomial, $\Pr[E_2] \leq (d - \ell)/|S|$. Since after $x_1, \ldots, x_{n-1}$ are fixed in $\neg E_2$, we have that $p(a_1, \ldots, a_{n-1}, x_n)$ is a non-zero polynomial of degree ℓ, we have that $\Pr[E_1 | \neg E_2] \leq \ell/|S|$. Thus $\Pr[E_1] \leq d/|S|$. ■

Application: Algorithm for Perfect Matching

Given a bipartite graph G with n vertices on the left and n vertices on the right, a perfect matching in the graph is a set of n disjoint edges in the graph. Here we give a simple randomized algorithm for computing whether or not a given graph contains a perfect matching.

Recall that the determinant of an $n \times n$ matrix M is defined to be

$$\mathsf{det}(M) = \sum_{\pi \in S_n} \mathsf{sign}(\pi) \prod_{i=1}^{n} M_{i\pi(i)},$$

where here S_n is the set of permutations on n elements, and $\mathsf{sign}(\pi)$ is either 1 or -1 depending on the permutation. We have algorithms for computing the determinant that run in time $O(n^3)$.

Now consider the matrix obtained from the input graph by setting

$$M_{ij} = \begin{cases} x_{ij} & \text{if } (i,j) \text{ is an edge,} \\ 0 & \text{otherwise.} \end{cases}$$

Then we have that $\mathsf{det}(M)$ is non-zero if and only if the graph has a perfect matching! Thus to test whether or not the graph has a perfect matching, it is enough to determine whether the polynomial $det(M)$ is non-zero or not. Observe that $\mathsf{det}(M)$ is a polynomial of degree at most n. Calculating this polynomial explicitly is too time consuming, since in general it may have an exponential number of monomials. Instead the following randomized algorithm works:

Input: A bipartite graph G with n vertices on each side.
Result: Whether or not G contains a perfect matching
For $i, j \in [n]$, sample a_{ij} uniformly at random from the set $\{1, 2, \ldots, 10n\}$;
Set

$$A_{ij} = \begin{cases} a_{ij} & \text{if } (i,j) \text{ is an edge,} \\ 0 & \text{otherwise ;} \end{cases}$$

if $\mathsf{det}(A) = 0$ **then**
  Output "No perfect matching";
else
  Output "There is a perfect matching";
end

Algorithm 1: Algorithm for deciding perfect matching

If the graph has no perfect matching, then clearly the polynomial $\mathsf{det}(M) = 0$, so the algorithm always outputs that there is no perfect matching. However, when the graph does contain a perfect matching, the probability that $\mathsf{det}(A) = 0$ is at most $1/10$ by the Schwartz-Zippel lemma.