Lecture 16

Today we will connect average-case complexity with worst-case complexity by introducing a #**P**-complete problem, PERMANENT, and showing that if PERMANENT is polynomial in the average case then it is polynomial in the worst case.

1 The PERMANENT Problem

The *permanent* is a number assigned to a matrix $M \in R$, for a ring R, according to the formula

$$\mathrm{perm}(M) = \sum_{\pi \in S_n} \prod_{i=1}^{n} M_{i,\pi(i)}$$

where S_n is the group of permutations on n items. The definition of the permanent closely resembles that of the determinant; the only difference is that the determinant has a coefficient $(-1)^{\mathrm{sgn}(\pi)}$ in front of the product. However, this resemblance is deceptive: the determinant can be calculated in polynomial time, whereas PERMANENT is #**P**-complete.

CLAIM 1
Let $R = \mathbb{Z}$ and let M be a 0/1 matrix, which we think of as representing a bipartite graph $G = (U, V, E)$ with $|U| = |V| = n$ as follows: number both edge sets U and V, separately, by $1, 2, \ldots n$, and let $M_{i,j}$ be 1 if edge $(i, j) \in E$, and 0 otherwise. Then $\mathrm{perm}(M)$ is the number of perfect matchings in G.

PROOF: Each $\pi \in S_n$ is a potential perfect matching, in that it can be thought of as a bijection between U and V. This corresponds to an actual perfect matching iff all edges $(i, \pi(i))$ are in E, i.e. if $M_{i,\pi(i)} = 1$ for all i, which is equivalent to saying that $\prod_{i=1}^{n} M_{i,\pi(i)} = 1$. Thus we get a summand of 1 in the formula for $\mathrm{perm}(M)$ for each perfect matching of G. $\square$

Suppose we allow the entries of M to be nonnegative integers. Then M represents a bipartite graph where the edges have multiplicity given by the entries of M. It still makes sense to interpret $\mathrm{perm}(M)$ as the number of permanent matchings, because the product of the multiplicities of the edges of a matching is the number of ways of realizing that matching.

Note that in $\mathbb{Z}_2$, $\mathrm{perm}(M) = \det(M)$ and thus is polynomial time. But for sufficiently large moduli (or no modulus at all) it's #**P**-complete.

THEOREM 2
Suppose for fixed n and $\mathbb{F}$, $|\mathbb{F}| > n^{O(1)}$, we can compute $\mathrm{perm}(M)$ on a fraction $1 - \frac{1}{2n}$ of the inputs M in polynomial time. Then there is a (worst-case) polynomial time probabilistic algorithm for computing $\mathrm{perm}(M)$.

Rather than proving this theorem directly, we will just observe that perm(M) is a degree n polynomial in indeterminates $M_{11}, M_{12}, \ldots, M_{1n}, \ldots, M_{nn}$, and prove the following much more general result.

Theorem 3
Suppose we have oracle access to a function $f : \mathbb{F}^n \to \mathbb{F}$ and f agreees with some degree d polynomial $p : \mathbb{F}^n \to \mathbb{F}$ on a fraction $1 - \frac{1}{2d}$ of its inputs where $|\mathbb{F}| \geq poly(n, d)$. Then we can compute $f(x)$ with high probability for all $x \in \mathbb{F}^n$ in $poly(n, d)$ time, assuming field operations can be performed in $poly(n, d)$ time.

Proof: Suppose that we input $x \in \mathbb{F}^n$ and pick $y \in \mathbb{F}^n$ uniformly at random. Consider the line $l(t) = x + ty$. Each point of l is uniformly distributed except $x = l(0)$. Now,

$$\begin{aligned} \Pr[p(l(i)) = f(l(i)) \text{ for } i = 1, 2, \ldots, d+1] &= 1 - Pr[\exists i \in 1, \ldots, d+1 : p(l(i)) \neq f(l(i))] \\ &\geq 1 - \frac{d+1}{2(d+1)} = \frac{1}{2} \end{aligned}$$

Now $q(t) = p(l(t))$ is a degree d polynomial in t, and we know $d+1$ of its values, so we can interpolate the others. Thus we have the following algorithm:

- input $x \in \mathbb{F}^n$
- pick random $y \in \mathbb{F}^n$
- consider $l(t) = x + ty$
- compute $f(l(1)), \ldots, f(l(d+1))$
- compute $q(t)$ of degree d such that $q(i) = f(l(i))$ for $i = 1, \ldots, d+1$
- output $q(0)$

This runs in poly(n, d) time, and with probability $\geq \frac{1}{2}$, it outputs $f(x)$. □

We can strengthen our result to hold even if $p(x)$ only agrees with $f(x)$ on $\frac{3}{4} + \epsilon$ of its inputs. This stronger proof depends on the following theorem, due to Berlekamp and Welch.

Theorem 4
Let $q : \mathbb{F} \to \mathbb{F}$ be a polynomial of degree d. Say we are given n pairs $(a_1, b_1), \ldots, (a_n, b_n)$ such that for fewer than $\frac{n-d}{2}$ pairs (a_i, b_i), $b_i \neq p(a_i)$. Then q can be reconstructed with $poly(n, d)$ field operations.

Intuitively, you should think of q as a message which we encode as the values $q(a_1), \ldots, q(a_n)$ as an encoding which uniquely specifices q if $n > d$. Say we receive $b_1, \ldots, b_n$. If we assume that fewer than $\frac{n-d}{2}$ of the transmitted values $q(a_i)$ have been been corrupted, then we can recover our original message q by attempting to extrapolate it from different sets of $d+1$ received values; a majority of them will give the correct q.

Now, as promised, we apply Berlekamp-Welch to strengthen our main theorem:

Theorem 5
Suppose we have oracle access to a function $f : \mathbb{F}^n \to \mathbb{F}$ and f agreees with some degree d polynomial $p : \mathbb{F}^n \to \mathbb{F}$ on a fraction $\frac{3}{4} + \epsilon$ of its inputs where $|\mathbb{F}| \geq poly(n, d)$. Then we can compute $f(x)$ with high probability for all $x \in \mathbb{F}^n$ in $poly(n, d)$ time, assuming field operations can be performed in $poly(n, d)$ time.

PROOF: Consider the following algorithm:

- input $x \in \mathbb{F}^n$
- pick random $y \in \mathbb{F}^n$
- consider $l(t) = x + ty$
- compute $f(l(1)), \ldots, f(l(d+1))$
- use Berlekamp-Welch to find q such that $q(t) = f(l(t))$ on more than $|\mathbb{F}| - \frac{|\mathbb{F}|-d}{2}$ values of t
- output $q(0)$

If $q(t) = p(l(t))$ and $f(l(t))$ agree on $> |\mathbb{F}| - 1 - \frac{|\mathbb{F}|-d-1}{2}$ values of t, Berlekamp-Welch will find $q(t) \equiv p(l(t))$ and the algorithm succeeds (and in polynomial time). So, assuming only $\frac{1}{4} - \epsilon$ of our points disagree, we need to show that only half of our runs fail.

$$\begin{aligned} \mathbf{E}\left[\#\text{ values of } t \in \mathbb{F} - \{0\} \text{ with } p(l(t)) \neq f(l(t))\right] &= \sum_{t \neq 0} \mathbf{E}\left[\begin{cases} 0 & \text{if } p(l(t)) = f(l(t)) \\ 1 & \textit{otherwise} \end{cases}\right] \\ &\leq (|\mathbb{F}| - 1)\left(\frac{1}{4} - \epsilon\right) \end{aligned}$$

So Markov tells us that

$$\Pr\left[\#t : p(l(t)) \neq f(l(t)) > \frac{n-d}{2}\right] \leq \frac{n(\frac{1}{4} - \epsilon)}{\frac{n-d}{2}} \leq \frac{1}{2} \quad \text{if } n \geq \tfrac{d}{2\epsilon}$$

Thus, our algorithm succeeds more than half the time, so it can be repeated to yield any degree of accuracy we wish. □

There is an even stronger result in which $\frac{3}{4} + \epsilon$ is replaced with $\frac{1}{2} + \epsilon$, but we won't prove it here.