# 30. Application of list decoding - Hardness of the Permanent (contd.)

## 30.1 Introduction

In this lecture we will study the permanent function and its computational complexity. We will use ideas from the decoding and list decoding of Reed Solomon codes to get results about the average case complexity of the permanent function.

Let $A$ be an $n \times n$ matrix, say over the integers. The function $\mathsf{PERM}_n(A)$ is defined as follows:

$$\mathsf{PERM}_n(A) = \sum_{\sigma \in S_n} \prod_{i \in [n]} A_{i,\sigma(i)}.$$

## 30.2 Permanent - average case to worst case

First, we will prove a result of Lipton that shows that if there is an algorithm that computes the permanent function for most matrices, then there is an algorithm that, with high probability, computes permanent for every matrix. Let $M_{n,p}$ denote the set of $n \times n$ matrices with entries from $\mathbb{F}_p$ for a large enough prime $p$. We will prove the following theorem.

**Theorem 30.1** (Lipton '91)**.** *Let $\mathcal{A}$ be an algorithm with the following property:*

$$\Pr_{B \in_r M_{n,p}}[\mathcal{A}(B) = \mathsf{PERM}_n(B)] \geq 1 - \frac{1}{3(n+1)}.$$

*Then, there is a randomized algorithm $\mathcal{A}'$ with the following property: For every $n \times n$ matrix $C$, we have*

$$\Pr[\mathcal{A}'(C) = \mathsf{PERM}_n(C)] \geq \frac{2}{3},$$

*where the probability is over the randomness of the algorithm $\mathcal{A}'$.*

*Proof.* Given a matrix $C \in M_{n,p}$, consider the matrix $M(x) = C + xB$ where $B \in M_{n,p}$. Then, $\mathsf{PERM}_n(M(x)) = \mathsf{PERM}_n(C+xB)$ is a polynomial of degree $n$ over $\mathbb{F}_p$. Let $B \in M_{n,p}$ be chosen at random. For each random $x \in \mathbb{F}_p$, the matrix $C + xB$ is distributed randomly in $M_{n,p}$. Now we choose $(n+1)$ random points $x_1, x_2, \ldots, x_{n+1}$ from $\mathbb{F}_p$. We use the algorithm $\mathcal{A}$ to compute $\mathsf{PERM}_n(C + x_iB)$. By the union bound, with probability at least $2/3$, $\mathcal{A}(C + x_iB) = \mathsf{PERM}_n(C + x_iB)$ for every $i \in [n+1]$. Now, we use the Langrangian interpolation to obtain the polynomial $\mathsf{PERM}_n(M(x))$ using the $n+1$ evaluations. To obtain $\mathsf{PERM}_n(C)$ we compute $\mathsf{PERM}_n(M(0))$. □

We now use the idea of the unique decoding of Reed Solomon codes to show a slightly stronger version of the theorem above. Let $M(x) = C + xB_1 + x^2B_2$. Notice that $\mathsf{PERM}_n(M(x))$ is an $2n$-degree polynomial. Suppose $B_1$ and $B_2$ are chosen u.a.r from $M_{n,p}$. Then for $x_1$ and $x_2 \in \mathbb{F}_p$, $M(x_1)$ and $M(x_2)$ are distributed randomly over $M_{n,p}$ and pairwise independent. First we show the following.

**Lemma 30.2.** *For $x_1 \neq x_2 \in F_p$, and $\alpha, \beta \in \mathbb{F}_p$, we have*

$$\Pr_{B_1,B_2 \in_r M_{n,p}}[M(x_1)[i,j] = \alpha \wedge M(x_2)[i,j] = \beta] = \frac{1}{p^2},$$

*where $M(x)[i,j]$ is the $(i,j)^{th}$ entry of the matrix.*

*Proof.* Now, for $i, j$, we have $C[i,j] + x_1B_1[i,j] + x_1^2B_2[i,j] = \alpha$. Therefore, we have $B_1[i,j] + x_1B_2[i,j] = x_1^{-1}(\alpha - C[i,j])$. Similarly, we have $B_2[i,j] + x_2B_2[i,j] = x_2^{-1}(\alpha - C[i,j])$. Solving for $B_1[i,j]$ and $B_2[i,j]$, the lemma follows. □

Let $Y_i$ denote the random variable that is 1 iff $\mathcal{A}(M(x_i)) = \mathsf{PERM}_n(M(x_i))$ where $x_i$ is chosen u.a.r from $\mathbb{F}_p$. From the previous lemma we know that $Y_i$s are pairwise independent. Suppose that $\mathcal{A}$ is an algorithm that computes the permanent correctly for at least $1/2 + \varepsilon$ fraction of the matrices in $M_{n,p}$. Then $\Pr[Y_i = 1] \geq 1/2 + \varepsilon$. If we perform evaluations at $N$ points $x_i$, then we get $N$ random variables $Y_1, Y_2, \ldots, Y_N$. Let $Y = \sum_i Y_i$ be the random variable that measures the number of times that algorithm $\mathcal{A}$ succeeds. Then $\mathbb{E}[Y] \geq \left(\frac{1}{2} + \varepsilon\right) N$.

We can think of this as the message obtained from an $[N, 2n+1, N-2n]_p$ RS code for a large enough $p$. Thus, we can correct up to $\frac{N}{2} - n$ errors.

By the Chebyshev inequality, we know that $\Pr[|Y - \mathbb{E}[Y]| > k] \leq N/k^2$. So, the probability that $\Pr[|Y - \mathbb{E}[Y]| > \beta n] < \frac{1}{\beta^2 N}$. For the decoding to give the correct polynomial, we want $Y \geq N/2 + n$. If we choose $\beta = \varepsilon/2$, then with probability at least $1 - \frac{4}{\varepsilon^2 N}$, we have $Y \geq \frac{N}{2} + \frac{\varepsilon}{2}N$. If we choose $N = \frac{12n}{\varepsilon^2}$, then with probability at least $2/3$, there are at most $N/2 - n$ points which are erroneous. We can then use the Welch-Berlekamp decoder to obtain the polynomial $\mathsf{PERM}_n(M(x))$. Then, as in the previous theorem, $\mathsf{PERM}_n(C) = \mathsf{PERM}_n(M(0))$.

So, similar to the theorem by Lipton, we have the following theorem due to Gemmell and Sudan.

**Theorem 30.3** (Gemmell-Sudan)**.** *Let $\mathcal{A}$ be an algorithm with the following property:*

$$\Pr_{B \in_r M_{n,p}}[\mathcal{A}(B) = \mathsf{PERM}_n(B)] \geq \frac{1}{2} + \varepsilon.$$

*Then, there is a randomized algorithm $\mathcal{A}'$ with the following property: For every $n \times n$ matrix $C$, we have*

$$\Pr[\mathcal{A}'(C) = \mathsf{PERM}_n(C)] \geq \frac{2}{3},$$

*where the probability is over the randomness of the algorithm $\mathcal{A}'$. Moreover, the running time of $\mathcal{A}'$ is poly$(n, 1/\varepsilon)$.*