Average case Complexity of the permanent

Now, we will prove a stronger result about the average-case complexity of the permanent.

Let us state the theorem, we want to prove:

Theorem: Suppose there is an algorithm A s.t

$$\Pr_{B \sim M_{n,p}} [A(B) = \text{Perm}(B)] \geq \frac{1}{\text{poly}(n)}$$

Then, there is an algorithm A' s.t $\forall B \in M_{n,p}$

$$\Pr [A'(B) = \text{Perm}(B)] \geq 2/3 .$$

Earlier, we saw how to go from $1/2 + \varepsilon$ to $2/3$ and the key idea was to do unique decoding of RS codes. Now, we are going to have even more errors. So we are going to do list decoding.

Let A be the matrix that is given, & $A^{i,j}$ denote the minors of A. Define the following matrix

$$D(x) = \sum_{i=1}^{n} \delta_i(x) A^{1,i} + \left[\prod_{i=1}^{n} (x-i)\right](B + xC)$$

where $\delta_i(x) = \begin{cases} 1 & \text{if } x = i \\ 0 & x \neq i \quad 1 \leq i \leq n \end{cases}$ we can find an $(n-1)$-degree polynomial $\delta_i(x)$

A few properties to note about D:

1. $B, C \in M_{n-1,p}$
2. for $i \in [n]$ $D(i) = A^{1,i}$
3. If we get $D(1), D(2), \ldots, D(n)$, then $\text{Perm}(A) = A[1,1] \cdot D(1) + \cdots + A[1,n] D(n)$

   Self reduction property of the permanent.

4. For B, C chosen at random from $M_{n,p}$ and $x \in_r \mathbb{F}_p$, the matrices $D(x)$ are pairwise independent.

⑤ $\text{Perm}(D(X))$ is a polynomial of degree at most n^2.

- Let q denote the success probability of the algorithm A.

Denote by Y_i the 0-1 random variable s.t

$$Y_i = \begin{cases} 1 & \text{if } A(D(n+i)) = \text{Perm}(D(n+i)) \\ 0 & \text{o/w} \end{cases}$$

Let $Y = \sum_{i=1}^{p-n} Y_i$

$$\mathbb{E}[Y] = (p-n)q$$

Since the Y_i s are pairwise independent, we have the following, due to the Chebyshev's inequality

$$\Pr\left[|Y - \mathbb{E}[Y]| > \frac{(p-n)q}{2}\right] \leq \frac{(p-n)4}{(p-n)^2q^2} = \frac{4}{(p-n)q^2}$$

$\therefore$ w.p $\geq 1 - \frac{4}{(p-n)q^2}$, $Y \geq \frac{(p-n)q}{2}$.

Thus we have a set of pairs $\{(\alpha_i, \beta_i)\}$ s.t $\exists$ a polynomial of degree $\leq n^2$ s.t it agrees with these pairs in at least $\frac{(p-n)q}{2}$ positions. Now if $\frac{(p-n)q}{2} \geq \sqrt{(p-n)n^2}$, then we can use the G-S algorithm to get a list of all such polynomials.

But there are two problems here:

① we need a good bound on the list size (we know that its polynomial).

② How do we get the correct polynomial from the list.

First we will solve ①: i.e. get a bound on the list of polynomials.

Let N denote the # of polynomials of degree $\leq n^2$ s.t they agree with $A(D)$ in $\geq \frac{(p-n)q}{2}$ positions.

Let $P_1, P_2, \ldots, P_N$ be these polynomials.

Let $S_i = \{j \mid P_i(j) = A(D(j))\}$

Obs 1: $|S_i| \geq \frac{(p-n)q}{2}$

Obs 2: For $i \neq j$ $|S_i \cap S_j| \leq n^2 - 1$

The reason is that two degree $\leq n^2$ polynomials can agree in at most n^2-1 positions.

$$\therefore \quad p-n \geq \left|\bigcup_i S_i\right| \geq \sum |S_i| - \sum |S_i \cap S_j|$$

$$\Rightarrow \quad p-n \geq \frac{N(p-n)q}{2} - \frac{N(N-1)}{2}(n^2-1)$$

$$\bigstar \quad p-n \geq \frac{N}{2}\left[(p-n)q - (N-1)(n^2-1)\right]$$

The constraints on p, q, N are as follows:

$$\frac{(p-n)q}{2} \geq \sqrt{(p-n)n^2} \Rightarrow (p-n) \geq \frac{4n^2}{q^2}$$

Substituting this in $\bigstar$ and simplifying, we get

$$N \leq \frac{4}{q}$$

So, now we have a small list of polynomials, but we are not sure which is the correct polynomial.

Observation: $\exists \alpha \in \mathbb{F}_p$ s.t $P_i(\alpha) \neq P_j(\alpha)$ $\forall i \neq j$ and P_i's obtained from the list decoding

algorithm.

Proof: $|\{i \mid P_i(\alpha) = P_j(\alpha)\}| \leq n^2$

$\therefore$ If $p > n^2 \binom{N}{2}$ then $\exists \alpha \in \mathbb{F}_p$

i.e. $p > n^2 N^2 \Rightarrow p > \frac{16n^2}{q^2}$

We are still not done. How do we know the actual value of $\text{Perm}(D(\alpha))$ to find out which of the ones are correct.?

Just recurse: Now we have an $(n-1)\times(n-1)$ matrix $D(\alpha)$ and we want to compute $\text{Perm}(D(\alpha))$.

So what is our success probability? First, the probability that we don't get sufficient good evaluations in $\leq \frac{1}{16n^2}$. And recursively, we need to do this for n steps. So the total error probability $\leq \frac{1}{16n}$.

So let us state the algorithm A' now

① Choose $B, C \in_r M_{n,p}$

② For $i \in [n+1, \ldots, p]$

$$\beta_i = A(D(i)) \quad \text{where}$$

$$D(x) = \sum_{i=1}^{n} \delta_i(x) A^{1,i} + \left[\prod_{i=1}^{n} (x-i)\right](B + xC)$$

(3) List decode $\{(i, \beta_i)\}$ to obtain a list L of polynomials of degree $\leq n^2$.

(4) Find $\alpha \in \mathbb{F}_p$ s.t $\forall P_i \neq P_j \in L$, $P_i(\alpha) \neq P_j(\alpha)$

(5) Recursively compute P^* s.t $P^*(\alpha) = \text{Perm}(D(\alpha))$

(6) Return $\text{Perm}(A) = \sum_{i=1}^{n} A[1,i] P^*(i)$.