6.842 Randomness and Computation April 2, 2014

Lecture 15

Lecturer: Ronitt Rubinfeld *Scribe: Luis Filipe Voloch*

1 Introduction

In this lecture we will go an algorithm for solving the following problem.

- **Input:** A binary function $f : \{-1,1\}^n \mapsto \{-1,1\}$ and a promise that there exists an $S^* \subset \{1,...,n\}$ such that $\hat{f}(S^*) > 0.5 + \epsilon$.
- **Output** A list L of subsets of $\{1,...n\}$ such that:
  1. For all $S \subset \{1,...n\}$ such that $\hat{f}(S) > 0.5 + \epsilon$, $S \in L$, and
  2. If $S \in L$, then $\hat{f}(S) > 0.5 + \epsilon/2$.
- **Requirements:** The algorithm must run in $poly(n, 1/\epsilon)$.

In order to get a feel for the problem, we will first cover two special cases. We will then provide an algorithm and a proof of correctness for the general case. Finally, we will note work done in a similar versions of the problem.

1.1 Historical Perspective

This question was first studied by Oded Goldreich and Leonid Levin [3] in the context of one-way-functions in Cryptography, and the algorithm presented in Section 4 is due to them.

2 Warm-up 1: ($\epsilon = 1/2$)

Let us first consider the case in which we have additional information about the Fourier concentration: that ϵ is 1/2. This means $\hat{f}(S^*) = 1$, which is equivalent to the case in which f is exactly a parity functions ($f = \chi_S$ for some S), and we must just find out exactly which indices i we should include in S. We can do so by checking by the following procedure. For each $i \in \{1,...n\}$:

- if $f(1) \neq f(e_i)$, then include i in S,
- if $f(1) = f(e_i)$, then do not include i in S.

Also, since we are doing only $n+1$ queries to the function, this works in $\theta(n) \in \text{poly}(n, 1/\epsilon)$. Note, that since f is a parity function, there are no other $S \in \{1,...,n\}$ that we should have outputted, since their Fourier coefficients are all $\hat{f}(S) = 0 < 1/2 + \epsilon/2 = 3/4$.

3 Warm-up 2: ($\epsilon > 1/4$)

Let us now consider the case in which $\epsilon = 1/4 + \delta$, for $\delta > 0$.

3.1 Algorithm

Consider the following algorithm: sample $s_i \in_R \{-1,1\}^n$, where $|\{s_i\}_i| = \frac{5}{2}n/\delta^2$. For each index $i \in \{1,...,n\}$:

- if, for the majority of the samples s_t, we have $f(s_t) \neq \chi_S(s_t)$ or $f(s_t e_i) \neq \chi_S(s_t e_i)$, then do not include i in S,
- otherwise include i in S.

3.2 Proof of Correctness

Lemma 1 *Consider the majority vote scheme described above. Then probability that the majority of the samples disagree with χ_S is less than $1/(10n)$.*

Proof Let $Z_j = \mathbb{1}\{f(s_j) \neq \chi_S(s_j)$ or $f(s_j e_i) \neq \chi_S(s_j e_i)\}$. Then, using the union bound, we have $\mathbb{E}Z_j \leq \mathbb{P}\{f(s_j) \neq \chi_S(s_j)\} + \mathbb{P}\{f(s_j e_i) \neq \chi_S(s_j e_i)\} = \frac{1}{2} - 2\delta$. In addition, since all Z_j are indicator random variables, we have $var(Z_j)) \leq 1$.

Now denote $Z = \sum_{j=1}^T Z_j$. Then $\mathbb{E}[Z] = \sum_{j=1}^T \mathbb{E}[Z_j] \leq T(\frac{1}{2} - 2\delta)$ and $var(Z) = \sum_{j=1}^T var(Z_j) \leq T$. We are now equipped to use Chebyshev inequality, where we get

$$\mathbb{P}(Z \geq T/2) \leq \mathbb{P}(|Z - \mathbb{E}(Z)| \geq 2t\delta) \leq \frac{var(Z)}{(2t\delta)^2} \leq \frac{t}{(2t\delta)^2} = \frac{1}{10n},$$

as desired. ■

Theorem 2 *The probability that any of the samples give is wrong is is less than $1/10$.*

Proof The A_i be the event that the we have included i incorrectly in the output S. Then we have

$$\mathbb{P}(error) = \mathbb{P}(\cup_{i=1}^n A_i),$$

where we can use the union bound and the Lemma above to get

$$\sum_{i=1}^n \mathbb{P}(A_i) = \frac{n}{10n} = 1/10.$$

■

Also note regarding the running time: for each of $i \in \{1,...,n\}$ we make $T = \theta(n/\delta^2)$ queries. Hence the running time is $\theta(nT) = \theta\left((n/\delta)^2\right)$ which is poly$(n, 1/\epsilon)$.

3.3 Note on an improvement

Note that in this case we could have just used the Chernoff bound instead of Chebyshev, since we are just dealing with a sum of Bernoulli random variables. This in turn would allow us to be have T be just $T = \theta(\log(n)/\delta^2)$. However, since in the following section we will Chebyshev, we decided to include that argument in this section.

4 General Case: $(\epsilon > 0)$

Note that the analysis done in the previous section breaks down if we have $\epsilon \leq 1/4$. In particular, where we take the union bound over the two events $\mathbb{P}\{f(s_j) \neq \chi_S(s_j)\} + \mathbb{P}\{f(s_j e_i) \neq \chi_S(s_j e_i)\} \leq 1/2$, which is of no use for our majority procedure.

Our fix to that is to choose the vectors $r_1, ..., r_T$ in a more clever way. In particular, instead of choosing them randomly as before, we will now pick them to be pairwise independent. We will do so by picking $s_1, ..., s_k \in \{-1, 1\}^n$, for $k = \log(T+1)$. These vectors will in turn generate a subspace of $2^k \approx T$ strings, where $T = 2n/\epsilon^2$.

4.1 Algorithm

We will modify our algorithm from the section above by selecting fewer samples our samples more cleverly.

Algorithm

- Choose $t = 2n/\epsilon^2$, and $k = \log(t)$.
- Choose vectors $s_1, ..., s_k \in \{-1, 1\}^n$
- For each of the $2^k \approx t$ assignments $(\sigma_1, ...\sigma_k) \in \{-1, 1\}^k$ (think of them as guesses to the values of $\chi_S(x)$):
  - For every $W \subset \{1, ..., K\}$
  - set $r_W = \bigoplus_{j \in W} s_j$, and $p_W = \prod_{j \in W} \gamma_k$
  - For all $i \in \{1, ..., n\}$, put i in $S_{\sigma_1,...,\sigma_k}$ if for the majority of p_W we have $p_W \neq f(r_w \bigodot e_i)$
  - Test if $S_{\sigma_1,...,\sigma_k}$ to check if more than $\frac{1}{2} + \frac{3}{4}\epsilon$ agrees with f on the majority of the output, and return $S_{\sigma_1,...,\sigma_k}$ only if so.

This generates many candidates for S. At the end, we can filter out the bad ones by testing if they agree with f on more than $\frac{1}{2} + \frac{3}{4}\epsilon$, and we will show that this filtering works via an analysis by Chernoff bound (Lemma 3).

4.2 Proof of Correctness

Lemma 3 *With high probability we can filter out all the members of the list S that agree with f on less than $\frac{1}{2} + \frac{1}{2}\epsilon$ of the inputs.*

Proof Consider the test in which we sample k random inputs $x_1, ...x_k \in \{-1, 1, \}^n$, and we filter out a candidate S if χ_S agrees with f on less than $\frac{1}{2} + \frac{3}{4}\epsilon$ of these k values. Let A_s be the event that S (with $\hat{f}(S) \leq \epsilon/2$) agrees with f on more than $\frac{1}{2} + \frac{3\epsilon}{4}$ fraction of the k inputs. Then by Chernoff bound we get

$$\mathbb{P}(A_S) \leq \exp\left(-(\frac{1}{4})^2 k/2\right) = \exp\left(-\theta(k)\right).$$

Now recall that there are less than $1/\epsilon^2$ candidate S in our list. Hence if we pick $k = \log(n/\epsilon^2)$ we get

$$\mathbb{P}(\cup_S A_s) \leq \sum_S \mathbb{P}(A_S) \leq \frac{1}{\epsilon^2} \exp\left(-\theta(k)\right) = \theta(\frac{1}{\epsilon^2}\frac{\epsilon^2}{n}) \to 0.$$

Furthermore, note that picking $k = \theta(\log(n/\epsilon^2))$ does not hurt us on the overall requirement on time complexity of the algorithm, which is still $\text{poly}(n, 1/\epsilon)$. ■

We will use an argument very similar to that of Lemma 1.

Theorem 4 *The algorithm above works with probability greater than* $1/2$.

Proof Let X_w denote the indicator random variable $\mathbb{1}\{p_w f(r_w \bigodot e_i(-1)^{\mathbb{1}\{i \in S\}}\}$. Then we see that the probability that the algorithm generates S when considering the guesses $S_{\sigma_1,...,\sigma_k}$ is exactly $\mathbb{E}[X_w]$. We will bound the probability of error by considering the mean and variance of this indicator random variable, and then using Chebyshev's inequality.

Its expectation is $\mathbb{E}X_w \geq \frac{1}{2} + \epsilon$, and the variance is $\sigma_w^2 = \mathbb{E}[X_w^2] - \mathbb{E}[X_w]^2 \geq \frac{1}{2} + \epsilon - (\frac{1}{2} + \epsilon)^2 = \frac{1}{4} - \epsilon^2$. Hence we can now apply the argument as in we did in Lemma 1. The analogous Lemma for here is:

$$\mathbb{P}(\sum_w X_w < \frac{t}{2}) \leq \frac{(\frac{1}{2})^2 - \epsilon^2}{t\epsilon^2} \leq \frac{1}{t\epsilon^2} \leq \frac{1}{2n}.$$

We can now use the union bound and we get that

$$\mathbb{P}(error) < n\frac{1}{2n} = \frac{1}{2},$$

as we wished. As our last step, by Lemma 3, we can with high probability filter all of the S that do not belong in the output. ■

5 Further Remarks

This question has also been studied in the context of random examples (as opposed to queries, as we did). In this case, Blum, Kannai, Wasserman provided the best known algorithm, with sub-exponential running time of $O(2^{n/\log n})$, in 2003 [1], where they used $2^{n/\log n}$ examples. In 2005, Vadim Lyubashevsky showed a different result [4], where the running time is worse, but requires substantially many examples. In particular, he showed that one can learn in time $O(2^{n/\log\log n})$ (which is worse than the 2003 result), but using only $\text{poly}(n)$ examples (which is better than the 2003 result). This question has also been studied under adversarial noise. In this case, the best known algorithm runs in $O(2^{n/\log n})$ and is due to Feldman, Golapan, Khot, and Ponnuswami in 2006 [2].

References

[1] Avrim Blum, Adam Kalai, and Hal Wasserman, "Noise-tolerant learning, the parity problem, and the statistical query model," JACM., vol. 50, iss. 4 2003 pp. 506-519.

[2] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Ponnuswami, "New Results for Learning Noisy Parities and Halfspaces," FOCS 2006.

[3] Oded Goldreich, Leonid Levin. "A Hardcore Predicate for All One-Way Function," Proceedings of STOC, 1989.

[4] Vadim Lyubashevsky, "The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem," APPROX'05/RANDOM'05.