6.895 Randomness and Computation February 22, 2006

Lecture 5

Lecturer: Ronitt Rubinfeld *Scribe: Josh Grochow*

Recall from last time, f is a Boolean function on Boolean strings, and we are attempting to find a linear function that approximates f by looking for large Fourier coefficients of $\hat{f}$, the Fourier transform of f. Throughout these notes, a "function" will always be a Boolean function on Boolean strings, unless otherwise specified.

$$f : \{\pm 1\}^n \to \{\pm 1\} \qquad \hat{f}(S) = \langle f, \chi_S \rangle = \frac{1}{2^n} \sum_{x \in \{\pm 1\}^n} f(x)\chi_S(x)$$
$$f(x) = \sum_{z \in \{\pm 1\}^n} \hat{f}(z)\chi_z(x) \quad \text{Parseval's: } \sum_S \hat{f}(S)^2 = 1 \text{ (for Boolean f)}$$

Definition 1 *A function $g : \{\pm 1\}^n \to \mathbb{R}$ is said to ε-**approximate** f in the L_2 norm if $E_x[(f(x) - g(x))^2] \le \varepsilon$.*

Claim 2 $$\underbrace{Pr[f(x) \neq sign(g(x))]}_{\textit{prediction error of sign}(g(x))} \le E_x[(f(x) - g(x))^2]$$

We proved this claim last time.

Suppose we are given query access to such a function f, and provided with a parameter θ. **Today's goal** is to examine an algorithm (the Goldreich-Levin algorithm) that will output all Fourier coefficients γ such that $|\hat{f}(\gamma)| \ge \theta$, and no coefficients γ with $|\hat{f}(\gamma)| < \theta/2$. (Coefficients whose magnitudes are between $\theta/2$ and θ may or may not be output.)

Before doing this, we will show that it is okay to ignore the small Fourier coefficients (i.e. those for which $|\hat{f}(\gamma)| < \theta/2$). The following lemma shows that if there is some function g which has few nonzero Fourier coefficients and ε-approximates f, then we can construct a function h which almost ε-approximates f and moreover has only large Fourier coefficients.

In fact, the Fourier coefficients output by the Goldreich-Levin algorithm will be exactly the Fourier coefficients of the h constructed in this lemma.

Lemma 3 *Given $f : \{\pm 1\}^n \to \{\pm 1\}$, suppose there exists g such that*

(G1) g is t-sparse (has no more than t nonzero Fourier coefficients)

(G2) g ε-approximates f (i.e. $E_x[(f(x) - g(x))^2] \le \varepsilon$).

Let T be the support of $\hat{g}$ (the set of nonzero Fourier coefficients of g). Then

$$h(x) = \sum_{z \in T : \hat{f}(z) > \varepsilon/t} \hat{f}(z)\chi_z(x)$$

has the following properties:

(H1) h is t-sparse

(H2) $E_x[(f(x) - h(x))^2] \le \varepsilon + O(\varepsilon^2)$

(H3) All nonzero coefficients of h are $> \varepsilon/t$.

Proof (H1) and (H3) are satisfied by construction. Since we defined h in terms of its Fourier representation, and the only nonzero coefficients of $\hat{h}$ are a subset of T, it is clear that h is t-sparse. Property (H3) is satisfied because h is constructed from only those coefficients which were greater than ε/t.

Since $E_x[(f(x) - g(x))^2] \le \varepsilon$, by assumption (G2), we need only show that

$$E_x[(f(x) - h(x))^2] - E_x[(f(x) - g(x))^2] \le O(\varepsilon^2).$$

To show this, we will take advantage of some basic properties of the Fourier transform (the map which sends a function k to $\hat{k}$). First, for any function k, we have $E[k^2] = \langle k, k\rangle$ which is the same as $\sum \hat{k}^2$, by Parseval's relation. Next, while we have not shown this in class, it is nonetheless the case that the Fourier transform is a linear operation: $\widehat{(f-g)}(z) = \hat{f}(z) - \hat{g}(z)$. So we have:

$$\begin{array}{rcll} E_x[(f(x)-g(x))^2] & = & \sum_z \widehat{(f-g)}^2 & \text{(by Parseval's relation)} \\ & = & \sum_z (\hat{f}(z)-\hat{g}(z))^2 & \text{(by linearity)} \\ & = & \sum_{z\in T}(\hat{f}(z)-\hat{g}(z))^2 + \sum_{z\notin T}(\hat{f}(z)-\hat{g}(z))^2 & \\ & = & \sum_{z\in T}(\hat{f}(z)-\hat{g}(z))^2 + \sum_{z\notin T}\hat{f}(z)^2 & (\hat{g}(z)=0 \text{ for } z\notin T) \end{array}$$

Where the last step follows because T is defined as the support of $\hat{g}$. Next, examine $E_x[(f(x)-h(x))^2]$:

$$\begin{array}{rcll} E_x[(f(x)-h(x))^2] & = & \sum_z (\hat{f}(z)-\hat{h}(z))^2 & \text{(by Parseval's and linearity)} \\ & = & \sum_{z\in T}(\hat{f}(z)-\hat{h}(z))^2 + \sum_{z\notin T}(\hat{f}(z)-\hat{h}(z))^2 & \\ & = & \sum_{z\in T}(\hat{f}(z)-\hat{h}(z))^2 + \sum_{z\notin T}\hat{f}(z)^2 & (\hat{h}(z)=0 \text{ for } z\notin T) \\ & = & \sum_{z\in T\wedge \hat{f}(z)\le\varepsilon/t}(\hat{f}(z)-\hat{h}(z))^2 + \sum_{z\notin T}\hat{f}(z)^2 & (\hat{h}(z)=\hat{f}(z) \text{ for } z\in T\wedge \hat{f}(z)>\varepsilon/t) \end{array}$$

Finally, subtracting the two expectations from one another, we get:

$$\begin{array}{rcl} E_x[(f(x)-h(x))^2] - E_x[(f(x)-g(x))^2] & = & \left(\sum_{z\in T\wedge \hat{f}(z)\le\varepsilon/t}\hat{f}(z)^2 + \sum_{z\notin T}\hat{f}(z)^2\right) - \left(\sum_{z\in T}(\hat{f}(z)-\hat{g}(z))^2 + \sum_{z\notin T}\hat{f}(z)^2\right) \\ & = & \sum_{z\in T\wedge \hat{f}(z)\le\varepsilon/t}\hat{f}(z)^2 - \sum_{z\in T}(\hat{f}(z)-\hat{g}(z))^2 \\ & \le & \sum_{z\in T\wedge \hat{f}(z)\le\varepsilon/t}\hat{f}(z)^2 \text{ (since } (\hat{f}-\hat{g})^2\ge 0) \\ & \le & t(\varepsilon/t)^2 = \varepsilon^2/t = O(\varepsilon^2) \end{array}$$

■

Now that we have shown that it is okay to approximate f by a function with only large Fourier coefficients, we must display an algorithm to actually *find* those coefficients. We begin by defining, for each partial Fourier coefficient $\alpha \in \{\pm1\}^k$, the function induced by choosing only Fourier coefficients that begin with α:

$$f_\alpha(x) = \sum_{\beta\in\{\pm1\}^{n-k}} \hat{f}(\alpha\beta)\chi_\beta(x)$$

One consequence of this definition is that for any fixed k, $f(x) = \sum_{\alpha\in\{\pm1\}^k} f_\alpha(x)$. Note that when $|\alpha| = n$, we have $f_\alpha(x) = \hat{f}(\alpha)$, and when $|\alpha| = 0$, we have $f_\alpha = f_\emptyset = f$.

The Goldreich-Levin algorithm, which we will denote by `coef`(α), is essentially a smart search for large Fourier coefficients (a binary search with efficient pruning of the search tree). To find the Fourier coefficients of f which are greater than θ, we simply call `coef`$(\emptyset)$. For the remainder of these notes, by large Fourier coefficients, we shall mean γ such that $|\hat{f}(\gamma)| \ge \theta$, and by small we shall mean $|\hat{f}(\gamma)| \le \theta/2$.

`coef`(α) begins by checking if f_α has any large Fourier coefficients at all (we'll see how the test is done shortly). If it does, then we partition the coefficients of f_α into $f_{\alpha(+1)}$ and $f_{\alpha(-1)}$ and recurse (where $\alpha(+1)$ denotes concatenating $+1$ to α).

```
Algorithm coef(α)
    if E[f_α^2] ≥ θ^2/2 then
        if |α| = n output α
        else output the union of coef(α(+1)), coef(α(−1))
```

For the moment, assume we can compute $E[f_\alpha^2]$ efficiently: we will see how to approximate this value at the end of the lecture. It now remains to show both that this algorithm is correct, and that it is efficient (probabilistic polynomial time).

The following lemma will bound the size of the search tree to be polynomial in $1/\theta$. Then, since we can approximate $E[f_\alpha^2]$ in polynomial time, this lemma shows that the entire algorithm runs in polynomial time.

Lemma 4 *1. Fewer than $1/\theta^2$ z's satisfy $|\hat{f}(z)|^2 \geq \theta$.*

2. For all $1 \leq k \leq n$, fewer than $1/\theta^2$ f_α's satisfy $E[f_\alpha^2] \geq \theta^2$.

Proof

1. Suppose more than $1/\theta^2$ z's satisfy $|\hat{f}(z)| \geq \theta$. Then for each of these, $|\hat{f}(z)|^2 \geq \theta^2$, so the sum of their squared magnitudes is greater than one. But this contradicts Parseval's theorem, which says that this sum should be exactly equal to one.

2. $E_x[f_\alpha^2(x)] = \sum_{\beta \in \{\pm 1\}^{n-k}} \hat{f}^2(\alpha\beta)$. If we then sum over all $\alpha \in \{\pm 1\}^k$, we get $\sum_{\alpha\beta \in \{\pm 1\}^n} \hat{f}^2(z)$, which is equal to one by Parseval's theorem. The same argument as above completes the proof.

■

Note that part (1) of the above lemma bounds the number of leaves that the Goldreich-Levin algorithm will explore, while part (2) bounds the number of visited nodes.

It turns out that these bounds are tight. Consider the function $f(x_1, \ldots, x_n) = maj(x_1, \ldots, x_k)$ for some $k < n$, where maj denotes the majority function. For all $S \subseteq [n]$ such that $S \cap \{k+1, \ldots, n\} \neq \emptyset$, $\hat{f}(S) = 0$, i.e. any Fourier coefficient related to the x_i for $i > k$ will be zero. The remaining Fourier coefficients will then be approximately $1/\sqrt{2^k}$, which must be approximately θ.

Next, we prove the correctness of the algorithm.

Lemma 5 *The Goldreich-Levin algorithm, as presented above, outputs all Fourier coefficients γ such that $|f(\gamma)| \geq \theta$, and no Fourier coefficients such that $|f(\gamma)| < \theta/2$.*

Proof Recall from part 2 of the above lemma that $E_x[f_\alpha^2] = \sum_\beta \hat{f}^2(\alpha\beta)$. Since $\hat{f}^2$ is positive for any argument, then if there is *any* β such that $|\hat{f}(\alpha\beta)| \geq \theta$, then $E_x[f_\alpha^2] \geq \theta^2$. So the algorithm explores all branches that have large Fourier coefficients, and when $|\alpha| = n$, outputs those coefficients.

Next, if every Fourier coefficient prefixed by α is small ($\leq \theta/2$), then $E[f_\alpha]^2 \leq \theta^2/4 < \theta^2/2$. In particular, if $|\alpha| = n$ and $|\hat{f}(\alpha)| \leq \theta/2$, α will *not* be output by the algorithm. ■

So the algorithm is both correct and efficient. All that remains is to show how to approximate $E[f_\alpha^2]$. If we had direct access to the Fourier coefficients of f, we could calculate this expectation directly by choosing only those coefficients prefixed by α. Since we only have black-box access to f, we need the following lemma.

While f_α is defined as a sum over all Fourier coefficients *pre*fixed by α, the following lemma lets us calculate $f_\alpha(x)$ by examing the values of f (not $\hat{f}$) that are *post*fixed by x—a consequence of the duality of the Fourier representation and the standard representation.

Lemma 6 *For all f, for all $1 \leq k \leq n$, for all $\alpha \in \{\pm 1\}^k$, and all $x \in \{\pm 1\}^{n-k}$,*

$$f_\alpha(x) = E_{y \in \{\pm 1\}^k}[f(yx)\chi_\alpha(y)].$$

Proof The Fourier representation of $f(yx)$ is $\sum_z \hat{f}(z)\chi_z(yx)$. Note that if $|y| = |z_1|$ and $z = z_1 z_2$, then $\chi_z(yx) = \chi_{z_1}(y)\chi_{z_2}(x)$. Plugging all of this back into the expectation over y, we get

$$\begin{aligned} E_y[f(yx)\chi_\alpha(y)] &= E_y\left[\left(\sum_{z_1,z_1} \hat{f}(z_1 z_2)\chi_{z_1}(y)\chi_{z_2}(x)\right)\chi_\alpha(y)\right] \\ &= \sum_{z_1,z_2} \hat{f}(z_1 z_2)\chi_{z_2}(x) E_y\left[\chi_{z_1}(y)\chi_\alpha(y)\right] \end{aligned}$$

By the orthonormality of the χ's, this last expectation is 0 if $z_1 \neq \alpha$ and 1 when $z_1 = \alpha$. This reduces the sum over z_1 to the single value $z_1 = \alpha$, so we are left with $\sum_{z_2} \hat{f}(\alpha z_2)\chi_{z_2}(x)$, which is exactly $f_\alpha(x)$. ■

This leads immediately to an algorithm for approximating $E_x[f_\alpha(x)^2]$: sample on x to get this expectation, and for each x, sample on y using the preceding lemma:

```
Algorithm approx(α)
    for all i = 1 to m_1
        choose x_i ∈_R {±1}^(n−k)
        for all 1 ≤ j ≤ m_2
            choose y_ij ∈_R {±1}^k
        let A_i = (1/m_2) Σ_{j=1}^{m_2} f(y_ij x_i) χ_α(y_ij)
    return B_α = (1/m_1) Σ_{i=1}^{m_1} A_i^2
```

Since we are only approximating $E[f_\alpha^2]$, this may lead to problems in our proof of correctness. Suppose all Fourier coefficients prefixed by some α are small. Then we know the true expectation of f_α^2 can be at most $\theta^2/4$. We can then choose m_1 and m_2 appropriately so that the probability of the approximation being greater than $\theta^2/2$ (i.e. the probability of a false positive) is exponentially small.

As mentioned, the Goldreich-Levin algorithm shows up in many different areas of research under various guises. It was discovered in cryptography, in the context of hardcore bits. The algorithm is also related to list decoding of linear codes, and to learning, which we will discuss in subsequent lectures.