CS 880: Advanced Complexity Theory 2/1/2008

Lecture 5: Dictatorship Testing

Instructor: Dieter van Melkebeek Scribe: Scott Diehl

Today we will finish developing a tester for dictator functions and then explore connections of this test (and the linearity test discussed last lecture) with error correcting codes and probabilistically checkable proofs.

1 Developing the Dictatorship Test

In the last lecture, we introduced and analyzed a local tester for linearity and began to reverse engineer how it could be modified to yield a test for dictatorships.

1.1 The Linearity Test

Recall that the linearity test operated by picking two points uniformly at random and then verifying that the function indeed satisfies linearity at these points.

- **Linearity test** T:
  - Pick $x, y \in \{-1, 1\}^n$ uniformly at random.
  - Set $z := x \cdot y$.
  - Accept if and only if $f(x)f(y) = f(z)$.

T easily satisfies the local tester property of making a constant number of queries— it makes three, to be exact. We must also show that it always accepts linear functions, and that the probability that it rejects any function f is at least the order of the distance of f from a linear function. More formally, we need to show that

$$\begin{aligned} (\forall f \in \text{LINEAR}) \Pr[T^f \text{ accepts}] &= 1, \text{ and} \\ (\forall f) \Pr[T^f \text{ rejects}] &\geq \Omega(D(f, \text{LINEAR})). \end{aligned}$$

Our analysis of T showed that if it accepts with high probability, then the given function f must have a large Fourier coefficient. In particular, we showed that

$$\begin{aligned} \Pr[T^f \text{ accepts}] &= \frac{1}{2} + \frac{1}{2} \sum_S (\hat{f}(S))^3 \qquad (1) \\ &\leq \frac{1}{2} + \frac{1}{2} \max_S \hat{f}(S). \end{aligned}$$

We have already seen that the existence of a large Fourier coefficient implies that f must be close to some linear function (namely, the character corresponding to the large Fourier coefficient), so we conclude the desired connection between acceptance probability and closeness to linearity. As a sanity check, notice that (1) agrees with the fact that T always accepts on linear f: A linear function's Fourier coefficients consist of one set to 1 and the rest to 0, so the right-hand side of (1) becomes $1/2 + 1/2 \cdot 1 = 1$ when f is linear.

1.2 Linear Junta Test

Our next step towards a test for dictatorships was to modify the linearity test T to arrive at a local test for linear juntas. We started with a high level idea to introduce a decreasing dependence of the acceptance probability on the size of the character with the heaviest Fourier coefficient. To do so, we inserted a factor of $\alpha^{|S|}$, for a parameter $0 < \alpha < 1$, into the right-hand side of (1), forming the condition

$$\begin{aligned}\Pr[T^f \text{ accepts}] &= \frac{1}{2} + \frac{1}{2}\sum_S \alpha^{|S|}(\hat{f}(S))^3 \qquad (2)\\ &\leq \frac{1}{2} + \frac{1}{2}\max_S \alpha^{|S|}\hat{f}(S).\end{aligned}$$

Since the term $\alpha^{|S|}$ becomes small when $|S|$ is large, a high acceptance probability now corresponds to the existence of a large Fourier coefficient corresponding to a small character— in other words, the function must be close to a small linear function. To find a local test that has this behavior, we used Fourier analysis to reverse engineer the required changes to the linearity tester T, and discovered that we must flip each bit of the string $z = x \cdot y$ with probability $\frac{1-\alpha}{2}$ prior to querying $f(z)$:

- **Linear junta test** T_α:
  - Pick $x, y \in \{-1, 1\}^n$ uniformly at random.
  - Set $z := x \cdot y$.
  - Flip each bit of z independently with probability $\frac{1-\alpha}{2}$.
  - Accept if and only if $f(x)f(y) = f(z)$.

In fact, by choosing α appropriately, we can guarantee that we accept (with high probability) only those functions that are close to characters of size at most one. This brings us very close to a test for dictatorships: Dictatorships are exactly the characters of size one, so the only remaining problematic case is when the function is close to the null character.

To achieve our "at-most-one" tester, we observe the behavior of T_α on characters of size at most one: T_α always accepts on the empty character, since the empty character is the constant 1 function and it is always the case that $f(x)f(y) = f(z)$; On a singleton character, the acceptance probability of T_α indicated by (2) is $\frac{1}{2} + \frac{\alpha}{2}$. Therefore, we have that:

$$\text{If } |S| \leq 1 \text{ then } \Pr[T_\alpha^{\chi_S} \text{ accepts}] \geq \frac{1}{2} + \frac{\alpha}{2}.$$

This is not quite what we desire, as our test has lost its perfect completeness; for that reason, this construction does not yield a local tester, although we can use it to construct a constant-query tester. To do so, we must analyze how to set α to achieve an appropriate soundness. Suppose that the test accepts with high probability, say

$$\Pr[T_\alpha^f \text{ rejects}] \leq \epsilon.$$

Then the relation (2) of acceptance to the Fourier coefficients and α tells us that

$$1 - \epsilon \leq \frac{1}{2} + \frac{1}{2}\max_S(\alpha^{|S|}\hat{f}(S)).$$

Rearranging the terms, we have that

$$1 - 2\epsilon \leq \max_S(\alpha^{|S|}\hat{f}(S)). \tag{3}$$

We would like to force this condition to correlate with f having a large Fourier coefficient on a character of size at most one. Notice that since both α and $\hat{f}(S)$ are less than one, the right-hand side of (3) is at most α^2 for S larger than a singleton. Thus, under the setting of

$$\alpha^2 < 1 - 2\epsilon,$$

the condition (3) must imply the existence of a large Fourier coefficient on a character of size at most one:

$$\exists S \text{ with } |S| \leq 1 \text{ s.t. } \hat{f}(S) \geq 1 - 2\epsilon.$$

Therefore, we can conclude that the distance from f to dictatorships or the constant 1 function is at most ϵ, provided $\alpha^2 < 1 - 2\epsilon$. This gives us the soundness condition:

$$(\forall f)\Pr[T_\alpha^f \text{ rejects}] \geq D(f, \text{DICTATORS} \cup \{1\}), \text{ provided } \alpha^2 < 1 - 2\epsilon.$$

As mentioned above, this does not quite yield a local test for dictatorships and the constant 1 function, as it lacks perfect completeness. It does, however, yield a tester by repeating the test many times to achieve the desired error parameter. This transformation is similar to the that from a local tester to a tester, but is slightly more complicated as we lack perfect completeness: we can no longer reject when *any* trial rejects, as this event can still occur when f has the desired property. If we ensure that T_α has some constant gap between acceptance and rejecting probabilities, we can instead count the proportion of accepting trials and accept if the proportion is on the positive side of the gap's midpoint. By the Chernoff bound, the number of trials required for this procedure to achieve a good probability of success is on the order of the inverse of the gap squared. We now calculate the conditions required for our test to have an acceptance gap along with the size of the gap: On the one hand, if the distance of f to DICTATORS $\cup\{1\}$ is at most ϵ, then the acceptance probability is at most $1 - \epsilon$, while a function that is actually a dictator or the constant 1 function is accepted with probability $\frac{1}{2} + \frac{\alpha}{2}$. Thus, we require that

$$1 - \epsilon < \frac{1}{2} + \frac{\alpha}{2}.$$

This condition can be simultaneously satisfied with our condition that α be small enough for the test to succeed, $\alpha^2 < 1 - 2\epsilon$, provided that

$$1 - 2\epsilon < \sqrt{1 - 2\epsilon},$$

which is satisfied provided that $\epsilon < \frac{1}{2}$. Furthermore, we compute that the gap is

$$\frac{1}{2} + \frac{1}{2}\sqrt{1-2\epsilon} - (1-\epsilon) \sim \epsilon/2,$$

which implies that our tester needs $O(\frac{1}{\epsilon^2})$ trials to succeed with probability $1-\epsilon$. Note that this compares to $O(\frac{1}{\epsilon})$ trials for the transformation from local tester to tester, with the local tester's perfect completeness accounting for the savings.

1.3 The Dictatorship Test

Our final step is to build upon the test for characters of size at most one to arrive at a test for dictatorships. Notice that if T_α passes, then f is close to either a dictatorship or the constant 1 function, so all that remains to to come up with some additional test to rule out the possibility of f being close to constant. One way to do this is to note that the constant function is always one, whereas a dictatorship is balanced. Although we are only guaranteed that f is *close* to one of these, it still suffices to simply query f at many points to determine if it is closer to all ones than it is to being balanced. We present a different way to rule out the constant function that involves some ideas that are useful later. The main idea is that if a function g is either a dictatorship or the constant function, then

$$g \in \text{DICTATORS} \Leftrightarrow g(-1,\ldots,-1) = -1. \tag{4}$$

This is because dictators output the value of a single variable, which is -1 for every variable in the case above, while the constant 1 function is always 1. Thus, our tester can check this condition to rule out the constant function; however, the given function f may only be close to a dictator, and therefore might be incorrect on the point $(-1,\ldots,-1)$, so we cannot test this condition directly. What we do know at this point is that f is close to a linear function, so we can use the self correctibility of linear functions to check the above condition with high probability. Namely, the procedure is:

- **Self-Correction Procedure to find** $g(x)$:
  - Pick y uniformly at random.
  - Return $f(x \cdot y)f(y)$.

Since y and $x \cdot y$ are individually distributed uniformly, each of $f(y)$ and $f(x \cdot y)$ differs from the correct g value with probability at most $D(f,g)$. The above procedure returns the wrong answer only if one of the queries to f is wrong, which by a union bound happens with probability at most $2D(f,g)$. Therefore, appending this procedure to our test, for $x = (-1,\ldots,-1)$, and accepting only if the returned value is -1, we successfully rule out the case where f is close to the null character with high probability.

We now derive the soundness condition yielded by the resulting dictatorship test. Suppose that the test rejects with probability ϵ. Then the first part of the test, T_α, must reject with probability at most ϵ. Therefore, T_α's soundness condition guarantees that f is ϵ-close to a dictator or the constant 1 function (provided that α is small enough). In the case where f is ϵ-close to the 1 function, notice that the self-correction test rejects with probability at least

$$1 - 2D(f,\{1\}) \geq 1 - 2\epsilon.$$

This quantity can be no more than the probability ϵ that the whole dictatorship test rejects:

$$\epsilon \geq 1 - 2\epsilon.$$

This constitutes a contradiction provided that $\epsilon < \frac{1}{3}$; for such values of ϵ, f cannot be ϵ-close to the 1 function, so it must be ϵ-close to a dictator, namely,

$$\Pr[\text{Dictator test rejects}] = \epsilon \geq D(f, \text{DICTATORS}).$$

This establishes the desired soundness for small ϵ; a dependence of $\Omega(D(f, \text{DICTATORS}))$ follows in any case for a constant smaller than 1/3 in the big-Omega notation.

We reiterate that although we have constructed a tester for dictators, we do not have a local tester since we do not have perfect completeness. Our tests definitely satisfy the $O(1)$ query requirement, though— the two procedures together yield 5 queries. This can be reduced to 3 queries by flipping a coin at the start to decide which subtest to run, at the cost reducing the soundness by a factor of two. In the homework we will see how to achieve a full-blown local test making only 3 queries.

In fact, once we have a 3-query local tester for dictatorships, we can get a 3-query local tester for every property $\mathcal{P}$ that is a subset of dictators: All that we need to do is modify the test that separates dictators from the constant function to also detect dictators that aren't in $\mathcal{P}$. To do so, we check the point x^* where $x_i^* = -1$ if and only if the i^{th} dictator function is in $\mathcal{P}$. Any dictator function in $\mathcal{P}$ outputs -1 on x^*, since x^* is -1 on the variables output by these functions, whereas the other dictator functions (and the constant 1 function) output 1. This allows us to produce the desired separation on $\mathcal{P}$ by accepting only when we compute that $f(x^*) = -1$ via self-correctability.

2 Connection to Coding Theory

We now discuss the connection of our tests to error correcting codes. The connection is simply stated: The linearity test is equivalent to a test for Hadamard codes, whereas the dictator test is equivalent to a test for the long code.

To see the connection of linearity testing to Hadamard codes, recall that the Hadamard code encodes an n-bit string a as the inner product of a with every n-bit string:

$$a \to (a \cdot x)_{x \in \{0,1\}^n}.$$

Another way to view this encoding is as the characteristic string of the character function χ_a, simply by the definition of a character as inner product of a subset of bit positions:

$$a \to \chi_a.$$

Therefore, closeness of a function to a character is equivalent to closeness of its characteristic string to a Hadamard codeword. Thus the linearity tester that we developed can be used to test for closeness to a Hadamard codeword. The local decodability propery of the Hadamard code then follows by using this test's closeness guarantee to allow recovery of the codeword via the self-correctability of linear functions.

Recall that the long code of a consists of *every* Boolean function evaluated at the point a, and thus has length 2^{2^n}:

$$a \to (f(a))_{f:\{0,1\}^n \to \{0,1\}}.$$

Another way to view this encoding is as the characteristic string of the a^{th} dictator function.

$$a \to \chi_{\{a\}}.$$

This connection holds because the domain of the a^{th} dictator function is the strings of length 2^n, which can be viewed as characteristic strings of n-bit functions. The a^{th} dictator function outputs the a^{th} bit of these strings, which is the same as the corresponding function evaluated at a. As with the Hadamard code, the tester for dictatorships yields a test for closeness to a codeword in the long code; building on this test with self-correction allows for local decoding of the long code.

3 Connection to Probabilistically Checkable Proofs

We conclude by discussing the connection of our testers to probabilistically checkable proofs (PCPs). In particular, the objects that we study are probabilistically checkable proofs of proximity (PCPPs), where the goal is to test whether a string is close to having some property when given access to a string that is purported to be able to aid us— a sort of "proof of proximity". The goal is to achieve this task using very few queries. More specifically, if the given string is good, then there should be a proof that leads to our acceptance, but if the string is ϵ-far from having the desired property, then every proof should be rejected with probability on the order of ϵ.

Definition 1. *A PCPP with proof length ℓ for a property $\mathcal{P} \subseteq \{0,1\}^N$ is an efficient randomized oracle machine T making $O(1)$ queries such that*

$$\begin{aligned} (\forall w \in \mathcal{P})(\exists \pi \in \{0,1\}^\ell) \qquad & \Pr[T^{w,\pi} \textit{ accepts}] = 1, \textit{ and} \\ (\forall w)(\forall \pi \in \{0,1\}^\ell) \qquad & \Pr[T^{w,\pi} \textit{ rejects}] \geq \Omega(D(w,\mathcal{P})). \end{aligned}$$

By efficient, we mean that T runs in time polynomial in $N + \ell$.

We now show how a local tester for dictatorships leads to a PCPP for any property, with doubly-exponential proof length. We point out that although its proof length is very long, this PCPP is highly nontrivial due to the fact that it makes only a constant number of queries.

Theorem 1. *For each $\mathcal{P} \subseteq \{0,1\}^N$, there exists a 3-query PCPP with proof length 2^{2^N}.*

Proof. The main idea behind this connection is that the proof π is supposed to be the long code of w. We've discussed how our dictatorship test allows us to test that π is close to the long code encoding of some string w' (with three queries). We can then use the subset dictatorship test we developed by modifying the dictatorship test, along with self-correction, to test that w' has property $\mathcal{P}$. If we pass these tests with high probability, we can surmise that there exists some w' with property $\mathcal{P}$ such that π is close to the long code of w'. All that remains is to verify that w is close to w', which we accomplish by running a test to compare random bit positions of w to those in w':

- Pick position i at random.
- Query w_i.
- Verify that $w'_i = w_i$.

If this test passes with high probability, then w must be close to w', and thus close to having property $\mathcal{P}$. The only hitch is that querying w'_i is nontrivial since π is only close to w''s long code; we can account for this fact by the self correction trick.

We now formally analyze the soundness condition. Suppose that the probability that the test rejects is ϵ. Then the properties of the long code and the subset dictatorship test imply that π is ϵ-close to the long code encoding of $w' \in \mathcal{P}$ for small enough ϵ. Furthermore, the comparison of w to w' rejects whenever a position i is selected where $w_i \neq w'_i$, unless the self correction procedure to query w'_i fails. The former event happens with probability $D(w, w') \geq D(w, \mathcal{P})$, while the latter event happens with probability at most 2ϵ due to π's closeness to the long code of w'. By a union bound, we have that

$$\epsilon \geq \Pr[\text{comparison fails}] \geq D(w, \mathcal{P}) - 2\epsilon.$$

This exhibits the desired linear dependence of the test rejecting on distance from $\mathcal{P}$: $\Pr[T^{w,\pi} \text{ rejects}] \geq D(w, \mathcal{P})/3$ whenever $D(w, \mathcal{P})$ is sufficiently small.

Finally, we point out that we can arrive at a test that makes only three queries by flipping coins to decide which of the above subtests to run. This reduces the number of queries made to the maximum of the subtests' queries, which can be verified to be three, at the cost of a constant-factor decrease in soundness. $\square$

We'll end by drawing a connection from our PCPP to the famous PCP theorem, and mention some further developments. Recall that the goal of the PCP theorem is to develop a PCP for satisfiability. We point out that our generic PCPP construction yields a PCP for satisfiability with similar parameters. To see this, consider the property $\mathcal{P}$ corresponding to the satisfying assignments of a given input formula φ on n variables with $m = |\varphi|$:

$$\mathcal{P} = \{w \in \{0,1\}^n | \varphi(w) = 1\}.$$

We construct a PCP that uses the PCPP's verifier T to check proofs consisting of a purported satisfying assignment w concatenated with the PCPP's proof π for closeness to $\mathcal{P}$. On a satisfiable φ, a satisfying assignment w concatenated with a good π—guaranteed to exist for such $w \in \mathcal{P}$—causes T to accept with probability 1. On unsatisfiable φ, $\mathcal{P}$ is empty, so the distance of any w to $\mathcal{P}$ can be considered maximal, namely, 1. Thus, the soundness property of the PCPP guarantees that we reject any proof with probability $\Omega(1)$, which meets our requirements of a PCP for satisfiability.

Our generic PCPP construction gives us a PCP for satisfiability with proof length length 2^{2^n}. By exploiting some structure inherent in NP-complete problems and using the Hadamard encoding of pairwise products of variables, we can reduce the length of the proof to size 2^{n^2}. The PCP theorem in its full effect yields PCPs with proofs of size poly(m) , and further work allows size $m \cdot \text{polylog}(m)$. Notice that the first two sizes depend on the number of variables n, whereas the latter two depend on the formula size m. It is possible for n to be much smaller than m in some cases, so we might prefer PCP's of size polynomial in n or even quasilinear in n. Unfortunately, recent results show that this is unlikely to happen— it would imply a collapse of the polynomial-time hierarchy.