Lecture 8: Boolean functions and degree of approximation

Rajat Mittal

IIT Kanpur

A *Boolean* function is a function from the domain $\{0,1\}^n$ to Boolean domain $\{0,1\}$ for some n. In other words, $f : \{0,1\}^n \to \{0,1\}$ is called a Boolean function. We will look at larger range, i.e., $f : \{0,1\}^n \to \mathbb{R}$. In these notes, all such functions will be called Boolean functions. The domain can be switched to $f : \{-1,1\}^n \to \mathbb{R}$ by the simple linear transform $\frac{1-x}{2}$ (we will mostly use this domain for Boolean functions). This will map 0 to 1 and 1 to -1.

Exercise 1. Why don't we identify 1 with 1 and 0 with -1?

These functions are of fundamental importance in many areas of computer science like chip/circuit design, logic, cryptography and complexity theory. As a student of computer science, you must already be familiar with gates like *AND*, *OR* and *NOT*. A familiar result is that any Boolean function can be computed using *NAND* gate (it is universal).

Our focus will be on the representation of Boolean functions as polynomials. The concept of representing or approximating Boolean functions by polynomials has been instrumental in many areas of computer science, like complexity theory, quantum computingi, learning theory and communication complexity. We will first see that every Boolean function can be represented as a polynomial. Then, we will define different measures of complexity of such functions depending upon the notion of representation of the Boolean function.

It turns out that one such very important measure, approximate degree, can be modeled as a linear program. We will look at the dual of this program. Finally, we will derive an explicit lower bound on the approximate degree of OR function using the dual of the linear program.

1 Representing Boolean functions

The easiest way to specify a Boolean function is by its truth table. That means, a function $f : \{-1,1\}^n \to \mathbb{R}$ is represented by 2^n real numbers.

In other words, a function $\{-1,1\}^n \to \mathbb{R}$ can be represented as a vector in vector space $\mathbb{R}^{2^n}$. We can define the usual inner product over this space,

$$\langle f|g\rangle = \sum_{x\in\{-1,1\}^n} f(x)g(x).$$

You already know a lot of examples of these Boolean functions. Notice that some of the functions have range as real numbers.

- *OR*: It is 0 if and only if all inputs are 1.
- *AND*: It is 1 if and only if all inputs are -1.
- *NOR*: It is 1 if and only if all inputs are 1.
- *PARITY*: Its range is $\{-1,1\}$. It is the multiplication of all the inputs, $PARITY(x) = x_1x_2\cdots x_n$.
- Constant function: Its value is $c \in \mathbb{R}$ for all inputs.

Exercise 2. What would be the Majority function?

Let us consider all functions of type $f : \{-1,1\}^n \to \mathbb{R}$. We saw that *PARITY* and constant function can be represented as a polynomial in variables. Is it possible to represent all Boolean functions as polynomials?

Exercise 3. Can you represent all functions $f : \mathbb{R} \to \mathbb{R}$ as polynomials? Notice that the domain is real numbers in this case.

The surprising answer for the Boolean domain is, YES!! The trick is called *interpolation*.

Interpolation: Let us take the simpler case of a function $f : \{0,1\}^n \to \mathbb{R}$ (notice the change in domain). First, polynomial representation of a very specific function.

Exercise 4. Can you find a polynomial which is 1 on all 1's input and 0 otherwise? What is this function called?

The answer is not very difficult and it is the polynomial $x_1 x_2 \cdots x_n$.

Note 1. It is the same polynomial as *PARITY*, but the domains are different.

The same technique can be generalized for an arbitrary input (not just all 1's input).

Exercise 5. Can you find a polynomial which is 1 on a particular input and 0 otherwise.

These polynomials are called *indicator polynomials*. Using these polynomials, every function $f : \{0,1\}^n \to \mathbb{R}$ can be written as a polynomial.

$$f(x) = \sum_{a \in \{0,1\}^n} f(a) \prod_{i:a_i=1} x_i \prod_{i:a_i=0} (1 - x_i).$$

Exercise 6. Verify that the polynomial defined above agrees with the function value on every input $x \in \{0,1\}^n$. Notice that the polynomial can take any input from $\mathbb{R}^n$.

Observe that these polynomials are of special kind, every variable appears at most once in any monomial. Such multivariate polynomials are called *multi-linear polynomials*. We have shown that every Boolean function can be represented as a multi-linear polynomial. Converse is trivially true. You will show in the assignment that this representation is unique.

Let $\mathbb{1}_a := \prod_{i:a_i=1} x_i \prod_{i:a_i=0} (1 - x_i)$ denote the indicator polynomial which is 1 at a and 0 otherwise. Then,

$$f(x) = \sum_{a \in \{0,1\}^n} f(a) \mathbb{1}_a.$$

1.1 Fourier representation

Our initial goal was to represent every function of the form $f : \{-1,1\}^n \to \mathbb{R}$ as a polynomial. We can apply the same trick, we only need to know the indicator polynomials for inputs $a \in \{-1,1\}^n$.

Exercise 7. What is the indicator polynomial for $a \in \{-1,1\}^n$?

Using the linear transformation to transfer $\{-1,1\}$ domain to $\{0,1\}$ domain,

$$\mathbb{1}_a = \prod_{i:a_1=1} \frac{1 + x_i}{2} \prod_{i:a_1=-1} \frac{1 - x_i}{2}.$$

Hence, any function $f : \{-1,1\}^n \to \mathbb{R}$ can be written as

$$f(x) = \sum_{a \in \{-1,1\}^n} f(a) \mathbb{1}_a.$$

Writing it in terms of monomials (notice that the degree of every variable is at most 1 in every variable),

$$f(x) = \sum_{s \in \{0,1\}^n} \hat{f}(s) \prod_{s_i=1} x_i.$$

A string $s \in \{0,1\}^n$ can be identified uniquely with a subset of $[n]$.

$$f(x) = \sum_{S \subseteq [n]} \hat{f}(S) \prod_{i \in S} x_i. \tag{1}$$

This is called a *Fourier representation* and $\hat{f}(S)$'s are called the Fourier coefficients of f. Actually, the partial parities, $\chi_S(x) = \prod_{i \in S} x_i$ for subsets $S \subseteq [n]$ are orthogonal to each other.

It can be proved by observing $\langle \chi_S(x) | \chi_T(x) \rangle = \sum_{x \in \{-1,1\}^n} \chi_{S \triangle T}(x)$. Here, $S \triangle T$ is the symmetric difference between two sets. The quantity $\sum_{x \in \{-1,1\}^n} \chi_S(x)$ is 0 except iff $S = \varphi$; you will prove this in the assignment.

This shows that functions $\chi_S(x)$ form an orthogonal basis of all multi-linear polynomials. So, Fourier representation is unique. Every function can be uniquely expressed in the form given by Eqn. 1.

Complexity theory focuses on finding whether a function is hard or easy under various complexity measures. Constant function seems to be one of the simplest functions. What measure can we choose to define complexity of a Boolean function?

Fourier degree of a function: One possible measure could be the degree of a function (also called *Fourier degree*). We can define degree of a monomial as sum of degrees of variables. Then, degree of a multivariate polynomial is the maximum degree of a monomial in its unique representation.

A function with low degree would be simple and high degree will be considered complicated.

Exercise 8. What is the maximum possible degree of a Boolean function on n variables?

Constant functions have 0 degree and are the simplest functions. Functions of kind $x_1, x_1 + x_2, x_1 + x_2 + \cdots + x_n$ are simple under this measure and have degree 1. Parity would be a hard function with maximum degree n.

It can be easily shown that the degree of most of the interesting functions is pretty high. One interesting class of functions is called symmetric functions, where the function value only depends on number of 1's in the input. It was shown by Gathen and Roche that all symmetric functions (except constant functions) have degree $n - O(n^{.525})$ [4]. We consider a slightly relaxed measure of degree.

2 Approximate degree

Motivated from approximation algorithm, we might want the polynomial to approximately represents f (instead of computing it exactly).

Exercise 9. What should be the meaning of a polynomial being an approximation of a Boolean function?

There can be multiple answers. One could be that the polynomial is equal to the value of function for a *significantly high* proportion of inputs. We will take a different approach (motivated from quantum query complexity), a polynomial p is an ϵ-approximation of f iff

$$|p(x) - f(x)| \leq \epsilon \quad \forall x \in \{-1,1\}^n.$$

Clearly, there can be multiple polynomials approximating f (why?). How should we define the complexity of a function in terms of its approximations?

Like the complexity of a problem is defined by the best algorithm for the problem, the *approximate degree* of a function f is defined as,

$$\widetilde{deg}_\epsilon(f) := \min_{p:\ p \ approximates \ f} deg(p).$$

It turns out that the exact value of ϵ does not matter, as long as it is a small constant. We can move from one ϵ to another by losing only a constant factor. A proof of this fact can be found in the paper by Avishay Tal in Appendix B [3]. For this note, we will not worry about the exact value of ϵ, you can safely assume $\epsilon = 1/3$.

This measure turns out to be pretty important in the study of Boolean functions and is polynomially related to decision tree complexity, quantum query complexity, sensitivity, usual degree etc. Lot of effort

has gone in giving bounds on approximate degree of various functions. We will see a linear programming approach towards characterizing approximate degree.

The linear program is going to be of a slightly different kind and will not directly output the value of approximate degree. Let us look for the best possible approximation of f (given) by a polynomial p of degree d.

Since we have fixed the degree, the Fourier coefficients of p can be treated as variables. The condition of approximability is,

$$|p(x) - f(x)| \leq \epsilon \quad \forall x \in \{-1,1\}^n.$$

Ignoring the absolute value, notice that the constraints are linear in our variables (f is known, p is linear in its coefficients).

Exercise 10. How will we get rid of the absolute value?

Getting rid of absolute values, we get a linear program (for a particular degree d) which finds the best possible approximation by a degree d polynomial.

$$\begin{aligned} & \min \quad \epsilon \\ \text{s.t. } & p(x) - f(x) \leq \epsilon \quad \forall x \in \{-1,1\}^n \\ & f(x) - p(x) \leq \epsilon \quad \forall x \in \{-1,1\}^n \\ & \epsilon \geq 0 \end{aligned}$$

The condition $\epsilon \geq 0$ does not affect our program (why?), but will help in simplifying the dual. Notice that there are $2 \cdot 2^n$ constraints in this linear program.

Exercise 11. How many variables are there?

If the value of this program is less than some constant ϵ, say $1/3$, then ϵ-approximate degree of f is less than d.

3 Dual witness

The above linear program is not of much use and just captures the definition of approximate degree. The real benefit is derived from its dual. Suppose $\phi_1(x)$'s are the dual variables for first set of equations and $\phi_2(x)$ for second set in the primal.

The dual is,

$$\begin{aligned} & \max \quad \langle f|\phi_1 - \phi_2\rangle \\ \text{s.t. } & -\sum_{x\in\{-1,1\}^n} \phi_1(x) + \phi_2(x) \leq 1 \\ & \langle \phi_1 - \phi_2|\chi_S\rangle = 0 \quad \forall S \subseteq [n] : |S| \leq d \\ & \phi_1(x), \phi_2(x) \leq 0 \quad \forall x \in \{-1,1\}^n \end{aligned}$$

Here, the first constraint corresponds to variable ϵ and second set of constraints to 2^n Fourier coefficients of p in the primal. Switching signs of ϕ_1, ϕ_2 (multiply them by -1),

$$\begin{aligned} & \max \quad \langle f|\phi_2 - \phi_1\rangle \\ \text{s.t. } & \sum_{x\in\{-1,1\}^n} \phi_1(x) + \phi_2(x) \leq 1 \\ & \langle \phi_2 - \phi_1|\chi_S\rangle = 0 \quad \forall S \subseteq [n] : |S| \leq d \\ & \phi_1(x), \phi_2(x) \geq 0 \quad \forall x \in \{-1,1\}^n \end{aligned}$$

Another trick will simplify the linear program considerably. Suppose, we reduce $\phi_1(x)$ and $\phi_2(x)$ by the same quantity, keeping them positive. It remains a feasible solution (first and third constraint are still satisfied and second is not affected) and the objective value does not change.

For every x, we reduce both till at least one becomes zero. Then, we can replace them by a single variable $\phi(x)$, it is $-\phi_1(x)$ if $\phi_2(x)$ becomes zero first, and it is $\phi_2(x)$ if $\phi_1(x)$ becomes zero first. In this case $\phi_1(x) + \phi_2(x)$ can be replaced by $|\phi(x)|$. This gives the linear program,

$$\begin{aligned} &\max \quad \langle f|\phi\rangle \\ \text{s.t.} &\sum_{x\in\{-1,1\}^n} |\phi(x)| \leq 1 \\ &\langle \phi|\chi_S\rangle = 0 \ \ \forall S \subseteq [n] : |S| \leq d \end{aligned}$$

The last constraint can be viewed in multiple ways. Since $\chi_S(x)$ are orthogonal to each other, it basically means that ϕ does not have a nonzero Fourier coefficient for any monomial of degree less than or equal to d. Such polynomials are called to have pure high degree d. Other way to view it is: $\langle \phi|g\rangle = 0$ for every polynomial of degree less than or equal to d (you will prove it in the assignment).

$$\begin{aligned} &\max \quad \langle f|\phi\rangle \\ \text{s.t.} &\sum_{x\in\{-1,1\}^n} |\phi(x)| \leq 1 \\ &\langle \phi|g\rangle = 0 \ \ (\textit{if } g \textit{ has degree } \leq d) \end{aligned}$$

Exercise 12. Show that we can renormalize ϕ, s.t., $\sum_x |\phi(x)| = 1$, without changing the objective value of the dual.

We get the final dual linear program,

$$\begin{aligned} &\max \quad \langle f|\phi\rangle \\ &\text{s.t. } \|\phi\|_1 = 1 \\ &\langle \phi|g\rangle = 0 \ \ (\textit{if } g \textit{ has degree } \leq d) \end{aligned}$$

The quantity, $\langle f|\phi\rangle$, is called the *correlation* of f with ϕ. A ϕ satisfying,

- $\|\phi\|_1 = 1$,
- correlation with f is greater than ϵ,
- pure high degree is d,

is called a *dual witness* for f.

Weak duality implies that a dual witness gives a lower bound of d on $\widetilde{deg}_\epsilon(f)$.

Actually it is not hard to show directly. Suppose, there is a dual witness ϕ,

$$\langle f - g|\phi\rangle \leq \left(\max_x |f(x) - g(x)|\right) \|\phi\|_1 \leq \epsilon \|\phi\|_1 = \epsilon.$$

But,

$$\langle f - g|\phi\rangle = \langle f|\phi\rangle - \langle g|\phi\rangle \geq \epsilon - 0 \geq \epsilon.$$

This gives a contradiction.

Why did we use linear programming then? We get a much stronger statement using strong duality. It tell us, if the approximate degree of f is d, we have a dual witness with pure high degree $d-1$.

Dual witness provides us an important technique to give lower bounds on the approximate degree of Boolean functions. Many lower bounds have been proven using dual witnesses. At this point, it seems to be the only way to prove lower bounds on non-symmetric functions. Let us see an example to bound the approximate degree of *OR* function.

3.1 Approximate degree of OR

It is known that a degree $O(\sqrt{n})$ polynomial approximates OR [1][Chapter 3.2] (follows even from quantum query complexity of OR, Grover's algorithm). We will give a matching lower bound of $O(\sqrt{n})$ using dual witness.

Remember the definition of $OR : \{-1, 1\}^n \to \mathbb{R}$ function. It is 0 if and only if all inputs are 1 (we map -1 to 1 and 1 to 0), and 1 otherwise. You might have seen OR defined over $\{0, 1\}$ domain, it does not matter. You will show in the assignment that the degree of OR for domain $\{-1, 1\}^n$ and for the domain $\{0, 1\}^n$ is the same.

We will actually lower bound the degree of NOR, it is 1 if and only if all inputs are 1, and 0 otherwise. It is not hard to see that the approximate degree of NOR and OR is same. Again, you will show a more general result in the assignment.

Suppose we want to show a lower bound of d on the degree of NOR. The dual witness ϕ should satisfy,

- pure high degree of ϕ is d,
- $\|\phi\|_1 = 1$,
- the correlation, $\langle NOR|\phi\rangle$, is bigger than some constant ϵ.

To tackle pure high degree, we use another trick. Multiplication by $PARITY$ switches pure high degree of a function to degree of a new function (assignment problem). So considering $\phi' = (\prod_i x_i)\phi$, we need,

- degree of ϕ' is at most $d' = n - d$,
- $\|\phi'\|_1 = 1$,
- the correlation, $\langle NOR|(\prod_i x_i)\phi'\rangle$ is bigger than some constant ϵ.

Exercise 13. Make sure that you can convert all conditions from ϕ to ϕ'.

The correlation is just $\phi'(1^n)$, because NOR is non-zero only at 1^n. The last two conditions, using normalization, are equivalent to the fact that $\frac{\phi'(1^n)}{\|\phi'\|_1}$ is bigger than some constant ϵ (why?). In other words, we need a low degree polynomial which has high value at 1^n and low value everywhere else (the norm is bounded).

Exercise 14. Can you think of any such function if the degree is not bounded?

Let us look at the function $h(x) = \sum_S \chi_S(x)$, we will show that it is perfect as a witness if degree is not a concern. The argument is very similar to the one which showed that $\sum_x \chi_S(x) = 0$ for any non-empty S.

For any $x \in \{-1, 1\}^n$, consider a coordinate, such that, $x_i = -1$ (we can tackle 1^n separately). Then, we can divide the subsets into pairs $S, S \cup \{i\}$, where S does not contain i. Their contributions cancel each other out. So,

$$h(x) = \begin{cases} 2^n \text{ for } x = 1^n \\ 0 \quad \text{otherwise} \end{cases}$$

So, a normalized $h(x)$ would have been a perfect ϕ', except for its degree.

Exercise 15. What is the degree of h.

How about truncating h to degree d'? Consider

$$h(x) = \sum_{S:|S|\leq d'} \chi_S(x).$$

It satisfies the degree condition, what are its correlation and norm? The new function h attains it maximum at 1^n and the values at other x's are small (cancellations). Are they small enough?

Define $\mathcal{H} := \{S : |S| \leq d'\}$, then $h(1^n) = |\mathcal{H}|$. The norm can be estimated by Cauchy-Schwarz inequality.

$$\|h\|_1 \leq 2^{n/2}\|h\|_2 = 2^{n/2}\sqrt{\sum_x \left(\sum_{S\in\mathcal{H}} \chi_S(x)\right)^2}. \tag{2}$$

Let us estimate the quantity $\sum_x(\sum_{S\in\mathcal{H}} \chi_S(x))^2$,

$$\begin{aligned}
\sum_x \left(\sum_{S\in\mathcal{H}} \chi_S(x)\right)^2 &= \sum_x \left(\sum_{S\in\mathcal{H}} \chi_S(x)\right)\left(\sum_{T\in\mathcal{H}} \chi_T(x)\right) \\
&= \sum_x \sum_{S,T\in\mathcal{H}} \chi_{S\triangle T}(x) \quad (S\triangle T) \textit{ is the symmetric difference} \\
&= \sum_{S,T\in\mathcal{H}} \sum_x \chi_{S\triangle T}(x) \\
&= 2^n \sum_{S,T\in\mathcal{H}, S=T} 1 \\
&= 2^n|\mathcal{H}|.
\end{aligned}$$

We get the equality,

$$\sum_x(\sum_{S\in\mathcal{H}} \chi_S(x))^2 = 2^n|\mathcal{H}|. \tag{3}$$

So the bound on $\frac{h(1^n)}{\|h\|_1}$ is $\frac{|\mathcal{H}|}{2^n\sqrt{|\mathcal{H}|}}$. Since $|\mathcal{H}|$ is bounded by 2^n, this is not constant. Unfortunately, $h(1^n)$ is not big enough as compared to other function values.

Note 2. We have used Cauchy-Schwarz to estimate the L-1 norm of h, that might be giving us a weaker bound.

We use another trick; in some sense, we square the function to make the peak more sharp (value at 1^n) but keep the degree fixed. Choose $\mathcal{H} := \{S : |S| \leq d'/2\}$ and define $h(x) = \left(\sum_{S\in\mathcal{H}} \chi_S(x)\right)^2$. The normalized version of h is going to be our ϕ' (h multiplied by $PARITY$ is our dual witness).

Exercise 16. Show that the degree of h is d' and $h(1^n) = |\mathcal{H}|^2$.

We need to estimate $\|h\|_1$, but that has already been done by Eqn. 3. So,

$$\frac{h(1^n)}{\|h\|_1} = \frac{|\mathcal{H}|^2}{2^n|\mathcal{H}|} = \frac{|\mathcal{H}|}{2^n}.$$

We get a combinatorial question. After what threshold, the number of subsets become a constant fraction of total number of subsets?

This is equivalent to estimating the sum of Bernoulli trials with success/fail probability 1/2. It is well known (central limit theorem), and $d'/2 = (n - O(\sqrt{n}))/2$ will already give us a constant fraction [2].

Exercise 17. Convince yourself that the normalized version of $(\prod_i x_i)h$ will be a dual witness of NOR for approximate degree $O(\sqrt{n})$.

4 Assignment

Exercise 18. Show that the degree of OR function is $\Theta(n)$.

Exercise 19. Given a function $f : \{0,1\}^n \to \mathbb{R}$, show that it can't have two different multi-linear representations.

Exercise 20. Show that $\sum_{x\in\{-1,1\}^n} \chi_S(x)$ is 2^n if $S = \varphi$ and 0 otherwise.

Exercise 21. What is the pure high degree of *PARITY* function on n bits?

Exercise 22. Show that $\langle f|g\rangle = 0$ for all g of degree d is equivalent to the condition that $\langle f|\chi_S\rangle = 0$ for all S such that $|S| \leq d$.

Exercise 23. Show that for any function f changing the domain from $\{-1,1\}^n$ to $\{0,1\}^n$ does not affect the degree.

Exercise 24. Show that the approximate degree of f and $c \pm f$ is same for any constant c using dual witness.

Exercise 25. Show that f has pure high degree d iff $(\prod_i x_i)f$ has degree $n - d$.

References

1. M. Bun and J. Thaler. Approximate degree in classical and quantum computing. `https://people.cs.georgetown.edu/jthaler/adegFnT.pdf`.
2. Stack Exchange. Lower bound for the sum of random variables. `https://math.stackexchange.com/questions/2540366/lower-bound-for-the-sum-of-bernoulli-random-variable`.
3. A. Tal. Shrinkage of de morgan formulae by spectral techniques. `https://eccc.weizmann.ac.il/report/2014/048/revision/1/download/`.
4. J. von Zur Gathen and J. Roche. Polynomials with two values. `https://link.springer.com/article/10.1007/BF01215917`.