Lecture 3: Basic concepts in Fourier analysis: influence and noise stability

Rajat Mittal

IIT Kanpur

The content of this chapter is inspired from the second chapter of the book *Analysis of Boolean functions* by Ryan OD́onell [2].

We have already mentioned that Boolean functions are useful in many diverse areas of computer science: complexity theory, social choice theory, cryptography and error correcting codes to take some examples. In many of these subjects, it is vital to consider how a function behaves when it is perturbed slightly. Fourier analysis is a very helpful tool and allows us to quantify this behavior.

In this lecture note, we will see two such perturbations. First, we will notice how the function value changes when we flip a particular coordinate of the input. This allows us to quantify the *impact* of a variable for the function. This is known as *influence*, we will define it in the next section and see an application. In the later half, the concept of *noise stability* will be looked at, where each coordinate is changed with some small probability.

1 Influence of a variable

Remember that $f : \{-1,1\}^n \to \{-1,1\}$ is a Boolean function. Given $x \in \{-1,1\}^n$, let $x^{\oplus i} \in \{-1,1\}^n$ denote the string where i-th bit is flipped. To take an example, $010^{\oplus 2}$ is the string 000. In a loose sense, a variable i has an impact in computing the function, if $f(x) \neq f(x^{\oplus i})$.

This motivates us to define *Influence* of a variable i,

$$Inf_i(f) = \Pr_x(f(x) \neq f(x^{\oplus i})).$$

Notice that the probability is over uniform distribution on the inputs x. In other words, the influence of a variable i is the number of x's such that the function value changes when i-th bit is flipped (divided by 2^n to normalize).

Exercise 1. What is the maximum possible value of $Inf_i(f)$.

Notice that influence is a way to capture the impact of a variable. One way to see it is: if the influence of a variable is 0, then the function does not depend on that variable (it could be safely skipped from the input).

Exercise 2. Prove the above claim.

Though, this quantification is not perfect. Consider the *dictator function* $f(x) = x_i$, we would assume that the impact of i on this function is very high. Still, both dictator on i and PARITY function have the same influence on the i-th variable ($Inf_i(f)$).

The total influence of a function is just the sum of influence of all variables.

$$Inf(f) = \sum_{i \in [n]} Inf_i(f).$$

Recall that $\hat{f}(S)$ denote the Fourier coefficient of f on set S. Our first task is to represent Inf in terms of the Fourier coefficients of f. Observe that the influence on the i-th variable can be written as,

$$Inf_i(f) = \frac{1}{2^n} \sum_x \left(\frac{\left(f(x) - f(x^{\oplus i})\right)}{2} \right)^2 .$$

Convince yourself that every term of the summation is 1 if and only if $f(x)$ is different from $f(x^{\oplus i})$. The key step is to convert the Fourier transform of $f(x)$ into the Fourier transform of $f(x^{\oplus i})$. This will require switching the sign of any monomial which has i in it.

Exercise 3. What is the Fourier representation of $f(x^{\oplus i})$.

$$Inf_i(f) = \frac{1}{2^n} \sum_x \left(\sum_{S:i\in S} \hat{f}(S)\chi_S(x) \right)^2 .$$

Let $g_i(x) = \sum_{S:i\in S} \hat{f}(S)\chi_S(x)$, using the generalization of Parseval's identity,

$$Inf_i(f) = \langle g_i | g_i \rangle = \sum_{S:i\in S} \hat{f}(S)^2.$$

Exercise 4. We have already seen that Fourier coefficients at the same level are equal for a symmetric function. Can you show now that influence is same for every variable?

Summing up, we get the expression for total influence of the function.

$$Inf(f) = \sum_S |S| \hat{f}(S)^2.$$

Observe that, if the Fourier coefficients are concentrated on higher degree terms, then the total influence will be higher.

1.1 Derivative of a Boolean function

The difference in function value when we switch one input bit, $f(x) - f(x^{\oplus i})$, is of such importance that we define a function for that. We will use a new notation here, $x^{i,1}$ means the i-th bit is set to 1. Formally, given a function $f : \{-1,1\}^n \to \mathbb{R}$, the derivative $D_i(f)$ (with respect to i-th bit) is defined as,

$$D_i(f)(x) = \frac{f(x^{i,1}) - f(x^{i,-1})}{2}.$$

Notice that the range of the function is generalized to real numbers. You might wonder why we have not taken the more symmetric version of the difference, $f(x) - f(x^{\oplus i})$, for defining the derivative. The reason is, we are viewing the range as real numbers, this definition allows us to write very similar formula for derivatives as real polynomials. The function $\frac{f(x)-f(x^{\oplus i})}{2}$ is known as the *Laplacian operator* and is denoted by $L_i(f)$, it is also useful in many contexts.

Exercise 5. Prove that $D_i(f+g) = D_i(f) + D_i(g)$, where $f+g(x) = f(x) + g(x)$.

You can also verify that $D_i(\chi_S) = \chi_{S/\{i\}}$ when S contains i and 0 otherwise. This gives the formula for $D_i(f)$ (similar to real polynomials),

$$D_i(f) = \sum_{S:i\in S} \hat{f}(S)\chi_{S/\{i\}}.$$

In other words, f can be written as,

$$f(x) = x_i D_i(f) + \sum_{S:i\notin S} \hat{f}(S)\chi_S.$$

The second term can be viewed as $\frac{f(x^{i,1})+f(x^{i,-1})}{2}$, which is the expectation over x_i. Define $E_i(f)(x) := \mathrm{E}_{x_i}[f(x)]$ (here everything except x_i is fixed). Then,

$$f = x_i D_i(f) + E_i(f).$$

The functions D_i, E_i does not depend upon i, this allows us to decompose f. This decomposition of f is pretty useful in proving properties of Boolean function using Induction on n.

Let us see some other benefits of $D_i(f)$. We can write Inf_i in terms of $D_i(f)$,

$$\mathit{Inf}_i(f) = \mathrm{E}_x[D_i(f(x))^2].$$

A function is called *monotone* if $f(x) \leq f(y)$ whenever $x \leq y$ coordinate-wise. The functions AND, OR and MAJORITY are monotone, but PARITY is not.

Exercise 6. Is Add_m a monotone function?

For the monotone function $D_i(f)$ is 1 if the function value switches by changing the i-th variable, otherwise it is 0. In other words, for a monotone f,

$$\mathit{Inf}_i(f) = \mathrm{E}_x[D_i(f(x))].$$

Noticing that the E_x is 0 for any χ_S except when S is empty.

$$\mathit{Inf}_i(f) = \hat{f}(\{i\}).$$

This gives the expression for total influence of a monotone function,

$$\mathit{Inf}(f) = \sum_i \hat{f}(\{i\}).$$

1.2 Degree of a Boolean function

Recall that we defined the degree of a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ as the size of the biggest subset S such that $\hat{f}(S) \neq 0$. This conforms with our understanding of degree of polynomials over reals.

Exercise 7. What is the degree of OR function?

You can easily verify that the degree of most of the functions we know (AND, OR, PARITY) is exactly equal to n.

Exercise 8. Can you find a Boolean function whose degree is not n?

After some thought, you can create such functions easily. The n-th degree coefficient only depends on the correlation with Parity. If the function is *balanced* with respect to parity (agrees and disagrees with parity at equal number of places), then it is of degree less than n.

On the other hand, can you find Boolean functions with degree 0 or 1? Sure, constant functions have degree 0 and dictator functions ($f(x) = x_i$ for some i) have degree 1. Though, this seems unfair. Dictator functions only depend on just 1 variable and can be thought of as a function over just 1 variable.

A more natural question is, what is the minimum degree of a Boolean function when it depends on all n variables?

Exercise 9. If the function is not Boolean (the range is $\mathbb{R}$), construct a function with degree 1 which depends on all n variables.

What does it mean for a function to depend on all n variables. Armed with influence, a function depend on all n variables iff $\mathit{Inf}_i(f) \neq 0$ for all i. We are now going to see the result of Nisan and Szegedy, where they show that any such function will have degree $\Omega(\log(n))$. Denote degree of a function f by $\deg(f)$.

Exercise 10. Can you show that $\deg(f) > 1$ if $n > 1$.

There are two main facts behind the proof. First, any small degree function is either identically zero or is non-zero with high probability. Secondly, degree of a function is bigger than the influence of the function.

Exercise 11. Why should these two ideas be enough, can you outline the proof assuming these two facts (think of the derivative of the function and its relation to influence).

The second fact is easy to prove from the formula for Influence in terms of Fourier coefficients. We know,

$$Inf(f) = \sum_S |S|\hat{f}(S)^2 \leq \deg(f)\left(\sum_S \hat{f}(S)^2\right).$$

Exercise 12. Prove $\deg(f) \geq Inf(f)$ for a Boolean function.

Let us formally state and prove the first idea.

Lemma 1. *Let $f : \{-1,1\}^n \to \mathbb{R}$ be a function with degree at most d. If f is not identically zero, then*

$$\Pr_x[f(x) \neq 0] \geq 2^{-d}.$$

Proof. We will prove the above statement by Induction on n.

Exercise 13. Prove the base case.

Without loss of generality, let x_n be one of the variables present in the Fourier expansion of f (it depends on x_n). Then $f = x_n D_n(f) + E_n(f)$, and x_n takes two possible values 1 and -1. The function will become $E_n(f) \pm D_n(f)$ depending on whether $x_n = \pm 1$.

Case 1: The function is not identically zero either when $x_n = 1$ or $x_n = -1$. In both these substitutions, restricted f will be a function of degree at most d and will be non-zero. Using induction, the probability that $D_n(f) \pm E_n(f)$ is nonzero is at least 2^{-d}. The input has $x_n = 1$ with half the probability. So,

$$\Pr_x[f(x) \neq 0] \geq 1/2(2^{-d}) + 1/2(2^{-d}) = 2^{-d}.$$

Case 2: One of $D_n(f) \pm E_n(f)$ is identically zero. Let us assume $D_n(f) - E_n(f) = 0$, other case will be similar. Since there is a unique Fourier representation for every function, $D_n(f)$ and $E_n(f)$ have the same Fourier representation. In other words $f = (x_n - 1)D_n(f)$. The degree of $D_n(f)$ is at most $d-1$ and hence it is nonzero with probability at least 2^{-d+1}. Considering the inputs when $x_n = 1$,

$$\Pr_x[f(x) \neq 0] \geq 1/2(2^{-d+1}) = 2^{-d}.$$

□

We are ready to prove the result of Nisan and Szegedy about the minimum degree required to represent a Boolean function.

Theorem 1 (Nisan and Szegedy [1]). *Let f be a Boolean function such that $\inf_i(f) \neq 0$ for all $i \in [n]$. Then, $\deg(f) \geq \frac{1}{2}\log(n)$.*

Proof. We have already done most of the hard work. Suppose the function has degree d. We know that $Inf_i(f) = \Pr(D_i(f) \neq 0)$. Since $D_i(f)$ is a polynomial of degree at most $d-1$, by Lemma 1, $Inf_i(f) \geq 2^{-d+1}$.

There are n variables and each influence is non-zero, this implies $Inf(f) \geq n2^{-d+1}$. Using the second idea, influence should be less than degree.

$$n2^{-d+1} \leq d$$

Taking log on both sides,

$$\log(n) - d + 1 \leq \log(d) \Rightarrow \quad d \geq \log(n) - \log(d).$$

If degree is more than $\log(n)$, there is nothing to prove. Otherwise $\log d \leq \log(\log(n))$. Giving us,

$$d \geq \log n - \log(\log(n)) \geq \frac{1}{2}\log(n).$$

□

We are able to prove that any *Boolean* function (which depends on all variables) will have degree at least $\log(n)$. Is this the tightest bound possible? It might be that all Boolean functions require degree at least $n/2$.

You will prove in the assignment that ADDR_m has degree $O(\log(n)$, where $n = m + 2^m$ is the input size of the function. This shows the tightness of the result proved.

The problem of minimum degree for a general Boolean function is solved. Notice that ADDR_m is a non-symmetric function. Can we say that degree of any symmetric Boolean function (non-constant) is very high? You might be surprised but it is easy to prove that any symmetric Boolean function has degree at least $n/2$. It was proved by Gathen and Roche [3] that degree of a symmetric function is almost full, it is more than $n - O(n^{.525})$. They actually conjectured that it is $n - O(1)$. This simple problem is still open. The best gap known is $n - 3$.

Given two functions f and g, what can you say about the degree of the composition of two functions? You will prove in the assignment that the degree of $f \cdot g$ is the multiplication of the degree of f and g.

2 Social choice theory

In social choice theory, we try to come up with rules to aggregate different opinions and come up with a collective output. One of the simplest example is a voting rule, where preference of voters is collected and a person (say) is selected. MAJORITY is the most natural voting rule, but we can also take MAJORITY of MAJORITY as a voting rule (Prime minister election in India). In general, any Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ can be viewed as a voting rule for an election with n voters and 2 candidates. The input and output bits can also be viewed as a YES/NO preference, instead of two candidates.

We have already seen voting rules like MAJORITY, AND (unanimous) and dictator (not very democratic). A few more examples are:

– Weighted majority: For a weight vector $w \in \mathbb{R}^n$, the function $\text{MAJORITY}_w(x)$ is the sign of $\sum_i w_i x_i$. It can be used when some voters' preferences are more important than others.

  Exercise 14. What is the weight vector for MAJORITY?

– Tribes: This is the function $\text{OR} \circ \text{AND}$. Think of votes coming from different tribes, the option is selected if any tribe unanimously selects the option. For example, a candidate is hired in the CSE department iff a sub-area (ML, Theory, Systems) unanimously selects the candidate.

What makes a voting rule more appropriate than others. Some of the properties of a good voting rule might be,

– Unanimous: If all voters choose -1 or if all of them choose 1, then the answer is -1 or 1 respectively.
– Monotone: If more voters choose a particular candidate (say -1) than the outcome should not switch from -1 to 1.

  Exercise 15. Convince yourself that this corresponds to the function being monotone.

– Odd: If we reverse the preferences, the outcome should be reversed. In other words, $f(-x) = -f(x)$.
– Symmetric: All voters are treated equally. One way to ensure this is to take f to be symmetric.

Making sure that the function is symmetric, is one way to claim that all voters are treated equally. There are weaker versions too, for instance, look at the definition of *transitive symmetric functions* in Ex. 37.

Exercise 16. Which of these good properties, of a voting rule, are not satisfied by MAJORITY or OR or AND.

Intuitively MAJORITY is the most natural choice for a voting rule. Let us see one way in which we can formalize this intuition. We will now show that majority maximizes number of agreement between winner and votes in its favor.

Let $f : \{-1,1\}^n \to \{-1,1\}$ be a voting rule. A good voting rule should be the one where voter's preferences agrees with the outcome of the election. For an input x, let w_x be the number of votes which agree with the function value $f(x)$. Ideally, we would like to maximize the expectation $\mathrm{E}[w_x]$.

If w_x votes agree with $f(x)$, then $n - w_x$ votes will have value $-f(x)$. This shows that $f(x)(\sum_i x_i) = 2w_x - n$. So,

$$\mathrm{E}[w_x] = \mathrm{E}[(n + f(x)(\sum_i x_i))/2] = n/2 + 1/2\mathrm{E}[f(x)(\sum_i x_i)] = n/2 + 1/2 \sum_i \hat{f}(\{i\}).$$

Exercise 17. What are we taking expectation over?

In other words, expected number of agreements only depend on $\sum_i \hat{f}(\{i\})$. Notice that this quantity is the total influence for a monotone function.

Let us see when this quantity is maximized.

$$\sum_i \hat{f}(\{i\}) = \mathrm{E}[f(x)(\sum_i x_i)] \leq \mathrm{E}[\left|\sum_i x_i\right|].$$

The last inequality is tight if and only if $f(x)$ has the same value as the sign of $\sum_i x_i$. In other words, MAJORITY maximizes the expected number of agreements among all Boolean functions. Again showing that it is a good choice for a voting rule.

2.1 Noise stability

In some scenarios, there might be a small possibility of an error while recording the vote. One good property of a voting rule might be, it is robust under such errors. In other words, the function value should not change (with high probability) under such small noise. Let us formalize this idea.

Let $x \in \{-1,1\}^n$ be a string or preferences of voters. The random variable $y \in \{-1,1\}^n$ is ρ-correlated with x if independently for all i,

- $x_i = y_i$ with probability $\frac{1}{2}(1+\rho)$.
- $x_i = -y_i$ with probability $\frac{1}{2}(1-\rho)$.

Fix our distribution on pairs (x, y), where x is chosen uniformly at random and y is ρ-correlated to x. For this section, let (x, y) denote the random variable chosen according to this distribution.

Exercise 18. Show that the probability of pair (x, y) is $\frac{1}{2^{2n}}(1+\rho)^{n-|x-y|)}(1-\rho)^{|x-y|}$. Remember that $|x-y|$ denotes the Hamming distance between x and y.

Intuitively for a robust voting rule, when x, y are ρ-correlated, the function value should not change (for a small ρ). We define *noise stability* to be,

$$Stab_\rho(f) = \mathrm{E}_{(x,y)}[f(x)f(y)].$$

Notice that the expectation is taken over all (x, y) pairs where these pairs are distributed according to the probability distribution defined above. We will skip the subscript on expectation when this probability distribution is used.

As before, our task would be to find a *Fourier* expression for noise stability. Let us compute the noise stability of partial parity.

$$Stab_\rho(\chi_S) = \mathrm{E}[\chi_S(x)\chi_S(y)].$$

Using independence of different coordinates i,

$$Stab_\rho(\chi_S) = \Pi_i \mathrm{E}[x_i y_i].$$

Exercise 19. Show that $\mathrm{E}[x_i y_i] = \rho$.

Using the exercise, we get,

$$Stab_\rho(\chi_S) = \rho^{|S|}.$$

The idea (like influence) would be to get the expression of noise stability by using linearity of expectation. You should try it, we will also need $\mathrm{E}[\chi_S(x)\chi_T(y)]$. A similar calculation can be done,

$$\mathrm{E}[\chi_S(x)\chi_T(y)] = \mathrm{E}[\chi_{S/T}(x)\chi_{T/S}(y)\chi_{S\cap T}(xy)].$$

Using independence of different coordinates i,

$$\mathrm{E}[\chi_S(x)\chi_T(y)] = \mathrm{E}[\chi_{S/T}(x)]\mathrm{E}[\chi_{T/S}(y)]\mathrm{E}[\chi_{S\cap T}(xy)].$$

Exercise 20. What is $\mathrm{E}[x_i]$ and $\mathrm{E}[y_i]$?

Since S, T are distinct,

$$\mathrm{E}[\chi_S(x)\chi_T(y)] = 0.$$

This allows us to compute the expression for noise stability in terms of Fourier coefficients,

$$Stab_\rho(f) = \sum_S \rho^{|S|}\hat{f}(S)^2 \qquad (1)$$

You will prove this formula in the assignment.

2.2 Arrow's theorem

Our task in this section is to prove the famous theorem of Arrow (1950) (a sightly weaker version) in social choice theory using the concept of noise stability. This Fourier analytic proof was given by Gil Kalai (2002). The content of this section are taken from the book by Ryan O'Donell.

Let us look at the problem first. We saw that MAJORITY is a very good choice as a voting rule for a 2-candidate election (last section). What happens when there are 3-candidates and n voters?

Exercise 21. Can you think of a aggregating strategy for 3 candidates given the voter preference?

Condercet came up with a natural way to generalize the voting rule (say some function f). We can hold pairwise elections between the three candidates, giving rise to 3 separate elections. Let $x \in \{-1,1\}^n$ be the preference of voters for the first 2 candidates, then $f(x)$ will be the winner for the first election; similarly, let y, z be the preference for other two elections and $f(y), f(z)$ be the outcome.

Let the three candidates be a, b, c. If we arrange x, y, z as columns, then every row will be a preference of a particular voter.

	x (a *vs* b)	y (b *vs* c)	z (c *vs* a)
v_1	1	-1	1
v_2	-1	1	-1
$\vdots$	$\vdots$	$\vdots$	$\vdots$
v_n	1	1	-1
winner	$f(x)$	$f(y)$	$f(z)$

To be consistent, every voter preference (row) should satisfy NAE_3 (not all equal) function. For example, a row of $1, 1, 1$ will reflect that the voter prefers a over b, b over c and c over a.

Exercise 22. Convince yourself that all satisfying inputs of NAE_3 function (6 out of 8) correspond to a valid voter preference.

Our task is to give a single winner. From the above exercise, if $f(x), f(y), f(z)$ is a satisfying input of NAE_3, then the candidates can be arranged in a sorted order. Such a winner is called a *Condercet* winner. Does such a winner always need to exist?

Exercise 23. Let $n = 3$ and f be the MAJORITY function; using symmetry, construct valid voter preferences where there is no Condercet winner. In other words, $f(x), f(y), f(z)$ are all equal.

This shows that MAJORITY might be a bad choice for a 3-candidate election. Though, majority is not alone. Arrow's theorem states that if you always want a Condercet winner, f should be a dictator function or a negated dictator function.

The Fourier analytic proof of Arrow's theorem calculates the probability of a Condercet winner, where the voter preference is uniformly distributed over all satisfying inputs of the NAE_3 function. We will show that this probability is 1 iff f is a dictator function.

Since NAE_3 is a $0,1$ function (or a random variable, if the inputs are random variable), the probability of having a Condercet winner for a function f is $\mathrm{E}_{(x,y,z)}[\mathrm{NAE}_3(f(x), f(y), f(z)]$. Here, each row of (x, y, z) is picked uniformly (and independently) from the satisfying inputs of NAE_3 function. Let us skip the subscript of expectation for the rest of the section.

Exercise 24. Why is the expected value not the constant term of NAE_3 function?

Let us expand the NAE_3 function in its Fourier representation and use linearity of expectation.

Exercise 25. What is the Fourier representation of NAE_3?

$$\mathrm{E}[\mathrm{NAE}_3(f(x), f(y), f(z)] = \frac{3}{4} - \frac{1}{4}\mathrm{E}[f(x)f(y)] - \frac{1}{4}\mathrm{E}[f(x)f(z)] - \frac{1}{4}\mathrm{E}[f(y)f(z)].$$

We want to calculate $E_{(x,y,z)}[f(x)f(y)]$ (other two terms will be same using symmetry). Looking at the inputs of NAE_3,

- $\mathrm{E}[x_i] = 0$, and similarly $\mathrm{E}[y_i] = \mathrm{E}[z_i] = 0$,
- the joint distribution of x and y is independent over i (notice that x_i is independent of other x_j's, prefrence of distinct voters),
- x_i, y_i is same with probability $1/3$ and different with probability $2/3$.

Exercise 26. Show that $\mathrm{E}[x_i y_i] = -1/3$.

In other words, we can assume that x, y are $-1/3$-correlated. So, $\mathrm{E}[f(x)f(y)] = \mathit{Stab}_{-1/3}(f)$. This gives the probability of Condercet winner to be,

$$\frac{3}{4} - \frac{3}{4}\mathit{Stab}_{-1/3}(f).$$

To prove the Arrow's theorem, we only need to check if this probability can be 1. This only happens when $\mathit{Stab}_{-1/3} = -1/3$. Remember the Fourier expression for noise stability, $\mathit{Stab}_{-1/3}(f) = \sum_S (-1/3)^{|S|}\hat{f}(S)^2$.

Exercise 27. Prove that $\mathit{Stab}_{-1/3}(f) = -1/3$ iff all the Fourier weight is on degree 1 monomials.

You will prove in the assignment below that all Fourier weight is on degree 1 monomials iff the function is a dictator or a negated dictator, proving Arrow's theorem.

3 Complexity of Boolean Functions in terms of Fourier Spectrum

In the previous section, we have studied various measures of complexity of boolean functions such as degree and total influence. Our first objective now is to characterize the complexity of boolean functions in terms of their Fourier spectrum. We recall that for boolean functions f having range $\{-1,1\}$, Parseval's identity implies that the sum over squared fourier coefficients is 1 i.e $\sum\limits_{S\subseteq[n]} \widehat{f}^2(S) = 1$. Thus $\widehat{f}^2(S)$ can be viewed as defining a discrete probability distribution over the characters or equivalently over the sets $S \subseteq [n]$. Using the above distribution, the complexity of a function f can be characterized in terms of the probability mass situated on high degree characters [2]:

Definition 1. *We say that the Fourier spectrum of $f\colon \{-1,1\}^n \to \mathbb{R}$ is ϵ-concentrated on degree up to k if*

$$\sum_{S\in[n],|S|>k} \hat{f}(S)^2 \leq \epsilon$$

The definition indicates that the name ϵ-concentrated is slightly unnatural, since the weight on terms upto degree k is actually $1-\epsilon$. For instance a function is 0-concentrated on degree up to k if and only if $\deg f \leq k$. Using a generalization of the earlier proof of the inequality $\deg f > \frac{\log n}{\log\log n}$, we can show that a function f satisfying $\deg f \leq d$ depends on atmost $d2^{d-1}$ variables.

The weight $1-\epsilon$ also corresponds to the probability mass over the characters upto degree k using the probability distribution defined above.

Recall that the degree of a boolean function only depends on the largest degree term having a non-zero fourier coefficient and disregards the size of the fourier coefficients. Thus the degree of a boolean function is discontinous in the sense that it be large even for functions having arbitrarily small coefficients for the highest degree terms.

Relation with Influence: Recall that the total influence of any function can by bounded by the degree i.e. $Inf(f) \leq \deg(f)$. Thus a function with a small degree must have small total influence. However, as can be seen from the example of the OR function, the converse is false. The OR function has degree n whereas the influence of each variable is $\frac{1}{2^{n-1}}$. Thus the total influence $\frac{n}{2^{n-1}}$ for the OR function is small, even though it has the maximum possible degree.

However, an application of Markov's inequality reveals that the total influence bounds the degree in an ϵ-concentrated sense. Concretely, observe that total influence $Inf(f)$ equals the expected degree of the function f under the probability distribution defined using the fourier coefficients i.e $Inf(f) = \sum_{S\in[n]} |S|\hat{f}^2(S)$. Thus applying Markov's inequality to the non-negative random variable $|S|$ yields:

$$\Pr_{S\sim S_f}[|S| \geq k] \leq \frac{Inf(f)}{k}$$

, where S_f is a random subset sampled with probabilities $\Pr[S_f = S] = \hat{f}^2(S)$. Choosing $k = \frac{Inf(f)}{\epsilon}$ gives:

$$\Pr_{S\sim S_f}[|S| \geq \frac{Inf(f)}{\epsilon}] \leq \epsilon.$$

Thus a function f with influence $Inf(f)$, is ϵ-concentrated upto degree $\frac{Inf(f)}{\epsilon}$.

Relaxation of ϵ-concentration: We can further generalize the notion of ϵ-concentration to arbitrary collections of subsets instead of upto k degree terms. We obtain the following notion:

Definition 2. *Let $\mathcal{F}$ be a collection of subsets $S \subseteq [n]$. We say that the Fourier spectrum of $f : \{-1,1\}^n \to \mathbb{R}$ is ϵ-concentrated on $\mathcal{F}$ if*

$$\sum_{\substack{S\subseteq[n]\\ S\notin\mathcal{F}}} \widehat{f}(S)^2 \leq \epsilon$$

Equivalently, we can say that the Fourier spectrum of $f : \{-1,1\}^n \to \mathbb{R}$ is ϵ-concentrated on $\mathcal{F}$ if $\Pr_{S \sim S_f}[S \notin \mathcal{F}] \le \epsilon$

In the following section, we will relate the above notion to the learnability of different classes of functions.

4 Extra reading: Learning Theory

Machine learning is an umbrella-term encapsulating algorithms that "learn" functions, probability distributions or tasks defined on an input space, using the values of the function on only a finite number of training points. The training points can be viewed as arising from an oracle. Thus the goal of machine learning is to learn the underlying functions using a minimum number of queries to the oracle.

Learning theory provides a framework for understanding which classes of functions, or hypotheses can be learned using a finite or a small number of training points.

For functions defined on the boolean hypercube, any function f can be uniquely identified through its values at the 2^n input points. However, as mentioned above, it is desirable to instead learn such functions "efficiently", i.e. using a minimum number of queries. For arbitrary functions, an adversary argument shows that learning without quering the values at all the points is infeasible. However, we will show that when functions are known to belong to a given a class of functions, denoted by **concept class**, learning can be made more efficient.

The function's value at the training points can be obtained in one of the following two ways:

1. Query model: The algorithm can choose the points where the function's values are obtained.
2. Random model: The points where the function's value is available are sampled randomly.

In this lecture, we will analyze the random model. We'll further assume that the points are sampled uniformly on $\{-1,1\}$. Concretely, we adopt the following notion of learning, known as Probably Approximately Correct (PAC) learning [2]:

Definition 3. *In the model of PAC ("Probably Approximately Correct") learning under the uniform distribution on $\{-1,1\}^n$, a learning problem is identified with a concept class $\mathcal{C}$, which is just a collection of functions $f : \{-1,1\}^n \to \{-1,1\}$. A learning algorithm A for $\mathcal{C}$ is a randomized algorithm which has limited access to an unknown target function $f \in \mathcal{C}$.*

In addition, A is given as input an accuracy parameter $\epsilon \in [0,1/2]$. The output of A is required to be (the circuit representation of) a hypothesis function $h : \{-1,1\}^n \to \{-1,1\}$. We say that A learns $\mathcal{C}$ with error ϵ if for any $f \in \mathcal{C}$, with high probability A outputs an h which satisfies $\Pr_x[f(x) \ne h(x)] \le \epsilon$.

For boolean valued functions f and h, $\Pr_x[f(x) \ne h(x)] \le \epsilon = \frac{1}{4}\|f-h\|_2^2$. Thus the above condition is equivalent to $\|f-h\|_2^2 \le 4\epsilon$. The quantity $\Pr_x[f(x) \ne h(x)]$ is called the generalization error or risk in machine learning. Thus minimizing $\Pr_x[f(x) \ne h(x)]$ is equivalent to minimization of $\|f-h\|_2^2$.

We further note that Parseval's identity implies that $\|h(x) - f(x)\|^2 = \left\|\hat{h}(x) - \hat{f}(x)\right\|^2$. Thus, if we can obtain a function h having fourier coefficients close to f, then h will also be close to f in squared norm over the uniform distribution defined on the inputs. As noted above, this is sufficient to ensure that the risk $\Pr_x[f(x) \ne h(x)]$ is small.

Thus our general strategy to learn a function f would be to obtain an h whose Fourier coefficients approximate those of f.

However, the function h, obtained by approximating the Fourier coefficients may not be Boolean valued. To ensure this, we instead use the function $\mathrm{sign}(h(x))$.

To approximate the Fourier coefficients of f, we rely on its values obtained at random samples of input points. The number of such samples, known as the query complexity will increase as the desired accuracy increases. Furthermore, the query complexity will also be large if a large number of Fourier coefficients need to be determined. Thus to allow estimation of most of the mass of Fourier coefficients, we need an efficient technique to approximate Fourier coefficients using a few samples as well as a restricted concept class of functions requiring the estimation of a only few coefficients. To summarize, it is desirable to satisfy the following two conditions:

1. Ability to estimate Fourier coefficients
2. The number of worthwile Fourier coefficients should be small.

In light of the above discussion, we restrict our concept class of functions to be ϵ-concentrated on some fixed collection of sets $\mathcal{F}$. We can then prove the following theorem:

Theorem 2. *Assume access to random examples from target function $f\colon\ : \{-1,1\}^n \to \{-1,1\}$. If we can identify a collection $\mathcal{F}$ of subsets such that f is $\frac{\epsilon}{2}$-concentrated on $\mathcal{F}$, then f can be learned in time poly$(|\mathcal{F}|, n, \frac{1}{\epsilon})$in addition to the time to identify $\mathcal{F}$.*

To prove the above theorem, we construct an algorithm that outputs, with high probability an h such that h is close to f.

We first make the following observation: Let f' be the restriction of f to fourier coefficients in $\mathcal{F}$ i.e $f' = \sum_{S\in\mathcal{F}} \hat{f}^2(S)$. Let g any function having all non-zero Fourier coefficients on the characters corresponding to sets in $\mathcal{F}$. We observe that:

$$\begin{aligned}\left\|\hat{g}(x) - \hat{f}(x)\right\|^2 &= \sum_{S\in[F]} (\hat{g}(x) - \hat{f}(x))^2 + \sum_{S\notin[F]} (\hat{f}(x))^2 \\ &= \sum_{S\in[F]} (\hat{g}(x) - \hat{f}(x))^2 + \frac{\epsilon}{2} \\ &= \|g - f'\|^2 + \frac{\epsilon}{2}\end{aligned}$$

Thus to ensure that $\|h - f\|^2 \le \epsilon$ it is sufficient to obtain an h satisfying $\|h - f'\|^2 \le \frac{\epsilon}{2}$.

Our final algorithm is as follows:

```
for $S \in \mathcal{F}$ do
    Obtain an estimate of $\hat{f}(S)$ using $t$ samples of $f(x)$
    $\hat{h}(S) \leftarrow \frac{1}{t}(\sum_{i=1}^{t} f(x_t)\chi_S(x_t))$
end
$h(x) \leftarrow \sum_{S\in\mathcal{F}} \hat{h}(S)(\chi_S(x))$
Output sign($h(x)$)
```

To prove that the fourier coefficients of $h(x)$ are close to those of f with high probability, we utilize the Chernoff bound, i.e. by applying Markov's inequality to e^{tX} where X is a random variable.

Theorem 3. ***Hoeffding's inequality for general bounded random variables**: Let $X_1, \ldots, X_N$ be independent random variables. Assume that $X_i \in [m_i, M_i]$ for every i. Then, for any $t > 0$, we have*

$$\mathbb{P}\left\{\sum_{i=1}^{N} (X_i - \mathbb{E}X_i) \ge \epsilon'\right\} \le \exp\left(-\frac{2\epsilon'^2}{\sum_{i=1}^{N} (M_i - m_i)^2}\right)$$

Applying the above inequality to $-X_i$ yields $\mathbb{P}\left\{\sum_{i=1}^{N} (X_i - \mathbb{E}X_i) \le -\epsilon'\right\} \le \exp\left(-\frac{2\epsilon'^2}{\sum_{i=1}^{N}(M_i - m_i)^2}\right)$. Thus using a union bound, we have the following inequality on the absolute deviation from mean:

$$\mathbb{P}\left\{\left|\sum_{i=1}^{N} (X_i - \mathbb{E}X_i)\right| \ge \epsilon'\right\} \le 2\exp\left(-\frac{2\epsilon'^2}{\sum_{i=1}^{N} (M_i - m_i)^2}\right)$$

We note that, for any set $S \in \mathcal{F}$, the random variable $\hat{h}(S) = \frac{1}{t}(\sum_{i=1}^{t} f(x_t)\chi_S(x_t))$ is the sum of t independent random variables $\frac{f(x_t)\chi_S(x_t)}{t}$ lying in the range $[-\frac{1}{t}, \frac{1}{t}]$ and having expectation $\hat{f}(S) = \frac{1}{t}\mathrm{E}_x[f(x)\chi_s(X)]$.

Therefore, by applying Hoeffding's inequality to $\hat{h}(S)$ we obtain:

$$\mathbb{P}\left\{\left|\hat{h}(S)-\hat{f}(S)\right| \geq \epsilon'\right\} = \mathbb{P}\left\{\left|\frac{1}{t}(\sum_{i=1}^{t} f(x_t)\chi_S(x_t)) - \hat{f}(S)\right| \geq \epsilon'\right\} \leq 2\exp\left(-\frac{\epsilon'^2 t}{2}\right)$$

Thus to ensure that $\left|\hat{h}(S)-\hat{f}(S)\right| \leq \epsilon$, with probability greater than $1-\delta'$, it is sufficient to have $2\exp\left(-\frac{\epsilon'^2 t}{2}\right) \leq \delta'$ or equivalently to sample $t \geq 2\log\frac{2}{\delta'}\frac{1}{\epsilon'^2}$ points.

Now, by choosing $\epsilon' = \sqrt{\frac{\epsilon}{2|\mathcal{F}|}}$ and $\delta' = \frac{\delta}{|\mathcal{F}|}$, and applying a union bound over all subsets $S \in \mathcal{F}$, we obtain that with probability greater than $1 - |\mathcal{F}| \times \frac{\delta}{|\mathcal{F}|} = 1-\delta$, we have $\|h(x) - f'(x)\|^2 = \sum_{S\in\mathcal{F}}(\hat{h}(S)-\hat{f}(S))^2 \leq |\mathcal{F}| \times \frac{\epsilon}{2|\mathcal{F}|} = \frac{\epsilon}{2}$. Since the number of queries, given by $|\mathcal{F}|t$ is $poly(|\mathcal{F}|, n, \frac{1}{\epsilon})$, this proves our theorem.

5 Two representations for Boolean functions

We have already hinted that a Boolean function f can be thought of as a function from $\{0,1\}^n \to \{0,1\}$ or $\{-1,1\}^n \to \{-1,1\}$.

Exercise 28. The 0 in one representation is mapped to 1 or -1? Why?

The two representations are equivalent and we can easily convert from one to another. For example, given the function as a polynomial from $\{0,1\}^n \to \{0,1\}$, we can write it in ±1 representation by changing the variables (say z) to $(1-z)/2$. The final function (output) becomes $1-2f$.

Exercise 29. Consider $f(x) = x_1 + x_2 - x_1x_2$, is it a Boolean function in $\{0,1\}$ domain? What is its representation in $\{-1,1\}$ domain?

Intuitively, it is also clear what is meant by this transformation. In the input as well as output the 0 is mapped to 1 (as identity) and 1 is mapped to -1. To take an example, OR function is 0 if and only if all inputs are 0. In ±1 representation, it is 1 iff all inputs are 1.

Exercise 30. What is the Parity function in 0/1 representation?

Why do we need two representations? As in other areas of mathematics, the two perspectives give different directions to look at any problem related to functions. In some cases, both representations are equally easy/difficult to analyze. In some cases, one representation is considerably easier to handle as compared to the other one. To take a few small examples, Parity is easier to represent in the ±1 domain; on the other hand AND is easier to write down in the 0/1 domain.

We saw that Fourier analysis was done in ±1 domain. Notice that Fourier analysis works for functions of the kind $\{-1,1\}^n \to \mathbb{R}$. So, sometimes we can even represent Boolean functions as $\{-1,1\}^n \to \{0,1\}$, and still use Fourier analysis. Which representation to use depends upon the context and the problem we are dealing with.

A helpful fact is, the polynomial degree of a Boolean function remains the same in the two representations. This is not direct, but requires a small proof. We will show that the degree in ±1 world is at most the degree in 0/1 world. A similar argument can be done in the opposite direction too, proving that the two degrees are equal.

Proof. Let f be a Boolean function. Suppose $p_{0/1}$ is the unique multilinear polynomial representation of the function f in 0/1 domain (NOT the Fourier domain). We have already shown that this polynomial representation is unique. If it is not clear to you, try proving it.

To convert f into ±1 domain, we just replace every input variable x_i with $(1-x_i)/2$. To convert the range, we take $p_{0/1}$ to $1-2p_{0/1}$. In other words,

$$p_{\pm}(x) = 1 - 2p_{0/1}\left((1-x_1)/2, (1-x_2)/2, \cdots, (1-x_n)/2\right).$$

Notice that this transformation is linear in input variables as well as the output. So the degree of $p_{\pm}$ is at most the degree of $p_{0/1}$.

Exercise 31. Why at most and not equal?

$\square$

To finish this section, we will prove a theorem, which is substantially easy to prove in the 0/1 domain.

Theorem 4. *Let f be a Boolean function of degree d (it is independent of the representation), then absolute value of any non-zero Fourier coefficient is more than $1/2^d$.*

The smallest absolute value of the non-zero Fourier coefficient is called the *min-entropy*, and has been studied widely. First try this exercise.

Exercise 32. Try to prove the theorem in the ± 1 world.

Proof. The proof turns out to be pretty straightforward in the 0/1 world. We already know that degree does not change in these two worlds.

The first observation is, all the coefficients in the 0/1 world are integers. This follows because the indicator functions for the 0/1 world have integer coefficients. So any function which takes only integer values will have integer coefficients.

To convert this polynomial into ± 1 domain, we replace every input variable z by $(1-z)/2$. Since every monomial has at most d input variables, every coefficient is a multiple of $1/2^d$. The transformation on range does not change this property. The theorem follows easily now.

Notice that we proved something stronger. We showed that if f is a Boolean function of degree d, then every Fourier coefficient is a multiple of $1/2^d$. $\square$

In the next few lectures, we are going to introduce many different complexity measures for these Boolean functions. The complexity of a Boolean function should not depend upon its representation (at least intuitively). We will see this in these lectures. We will keep changing the representation depending upon our convenience, whatever is easy to analyze.

6 Assignment

Exercise 33. What is the maximum possible *Inf* for a function. Is it possible to achieve this maximum value? What functions achieve this value?

Exercise 34. What is the influence of target and address variables in addressing function?

Exercise 35. Express influence of i-th variable as an expectation of D_i, and derive the expression for Influence in terms of Fourier coefficients.

Exercise 36. Let σ be a permutation on input bits. A function is invariant under σ if $f(x) = f(\sigma(x))$. If a function is invariant under σ, show that

$$\hat{f}(S) = \hat{f}(\sigma(S)).$$

Exercise 37. A group of permutations G is called *transitive*, if given a pair i, j there exists a permutation $\sigma \in G$ such that $\sigma(i) = j$. A function is called transitive symmetric if the function is invariant under the action of some transitive group G. Notice that a symmetric function is transitive symmetric under the complete group S_n.

Show that if a function is monotone and transitive symmetric, then $Inf_i(f)$ is less than $1/\sqrt{n}$ for all i.

Exercise 38. Prove the following:

$$\deg(f \cdot g) = \deg(f) \cdot \deg(g).$$

Exercise 39. Show that there exist infinite n's such that MAJORITY_n has full degree.

Exercise 40. Show that the degree of Add_m is $m+1$.

Exercise 41. Which of the good properties of a voting rule are not satisfied by weighted majority or tribes.

Exercise 42. Prove that the only Boolean function of degree 1 are dictator functions, negated dictator or constant functions.

Exercise 43. Prove Eq. 1.

Exercise 44. Given a function in ± 1 representation, how will you convert it into 0/1 representation?

References

1. N. Nisan and M. Szegedy. On the degree of boolean functions as real polynomials. *Computational Complexity, volume 4, pages 301–313*, 1994.
2. Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.
3. von Zur Gathen and J. Roche. Polynomials with two values. *Combinatorica 17*, pages 345–362, 1997.