CS 6817: Special Topics in Complexity Theory **Spring 2025**

Lecture 8: Feb 13, 2025

Lecturer: Eshan Chattopadhyay *Scribe: Yunya Zhao*

1 Condorcet elections

The 3-party condorcet election setting is as follows: 3 candidates, n voters, the voters each gives a linear ranking of the candidates, and every pair of candidates are compared according to the ranking, as shown in the table below.

	x	y	z
	$a(+1)b(-1)$	$b(+1)c(-1)$	$c(+1)a(-1)$
voter 1: $b > a > c$	-1	$+1$	-1
voter 2: $b > c > a$	-1	$+1$	$+1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
voter n: $a > b > c$	$+1$	$+1$	-1

A voting rule is a function $f : \{-1, 1\}^n \to \{-1, 1\}$. String x is a string in $\{-1, 1\}^n$ representing each voter's comparison between candidate a and b: $x_i = 1$ if voter i prefers candidate a and $x_i = -1$ otherwise. Candidate a wins over b if $f(x) = 1$ and vice versa. String y and z are defined similarly for the other two pairs of comparisons. The candidate a wins the whole election if a wins both pairwise comparisons involving a, that is $f(a) = 1$ and $f(z) = -1$.

Theorem 1.1 (Arrow's Theorem)**.** *Let the voting rule f be balanced and unanimous. Then the only such rule for which there is a condorcet winner is DICTATORS* $= \{x_1, \cdots, x_n\}$

To prove this, we first define the not-all-equal predicate to be NAE : $\{-1, , 1\}^n \to \{0, 1\}$, where

$$\text{NAE}(\omega) = \begin{cases} 0 & \text{if } \omega_1 = \omega_2 = \omega_3 \\ 1 & \text{otherwise} \end{cases}$$

There is a condorcet winner if and only if $\text{NAE}((f(x), f(y), f(z)) = 1$.

Observation 1.2. *For any voter, there are* $3! = 6$ *ways of ranking a, b, c. Observe that for any voter i, (x_i, y_i, z_i) cannot be $(+1, +1, +1)$ or $(-1, -1, -1)$ assuming they all give valid rankings.*

Observation 1.3. $NAE(x) = \frac{3}{4} - \frac{1}{4}x_1x_2 - \frac{1}{4}x_2x_3 - \frac{1}{4}x_3x_1$

Claim 1.4. *If each voter independently picks a ranking uniformly at random, then the probability that there exists a condorcet winner is* $\frac{3}{4}(1 - NS_{-\frac{1}{3}}(f))$.

With Claim 1.4, we can prove Arrow's Theorem.

Proof of Theorem 1.1. For there to always be condorcet winner, $\Pr[\exists \text{ a condorcet winner}] = 1$, that is, $NS_{-\frac{1}{3}}(f) = -\frac{1}{3}$. By a result from last lecture, this is true if and only if f is the dictator function. □

We now try to prove Claim 1.4.

Proof of Claim 1.4. Observe for any i, $x_i = y_i$ with probability 1/3 for a random ranking. This is saying we can write y as if sampled from the noisy distribution $\mathcal{N}_\rho(x)$ where $\rho = -1/3$. We can do similar for y, z and x, z.

$$\begin{aligned}\mathbb{E}[\text{NAE}(f(x), f(y), f(z))] &= \frac{3}{4} - \frac{1}{4}\mathbb{E}_{x\sim\{\pm1\}^n, y\sim\mathcal{N}_\rho(x)}[f(x)f(y)] - \frac{1}{4}\mathbb{E}[f(y)f(z)] - \frac{1}{4}\mathbb{E}[f(z)f(x)] \\ &= \frac{3}{4} - \frac{3}{4}NS_{-\frac{1}{3}}(f)\end{aligned}$$

The first equality follows from Observation 1.3 and the second equality is by definition of the noisy stability $NS_\rho(f)$. □

2 Fourier concentration of computational models

We now switch gears and start a new topic: the Fourier concentration of computational models. We first give a few definitions. Let $f : \{-1, 1\} \to \{-1, 1\}$. [1]

For $k = 0, 1, \cdots, n$, define the *level-k mass* to be

$$W^k[f] = \sum_{S:|S|=k} \hat{f}(S)^2$$

and define the *level k tail* to be

$$W^{\geq k}[f] = \sum_{S:|S|\geq k} \hat{f}(S)^2$$

Claim 2.1. *For any $f : \{-1, 1\}^n \to \{-1, 1\}$ and $\varepsilon > 0$, $W^{\geq k}[f] \leq \varepsilon$, for $k = \frac{I(f)}{\varepsilon}$.*

Proof. Recall the spectral distribution $\mathcal{S}$ on $2^{[n]}$ where $\Pr[\mathcal{S} = T] = \hat{f}(x)^2$, and $\mathbb{E}_{T\sim\mathcal{S}}[|T|] = I(f)$. Then by Markov's inequality,

$$W^{\geq k}(f) = \Pr_{T\sim\mathcal{S}}[|T| \geq k] \leq \frac{I(f)}{k} = \varepsilon$$

□

Corollary 2.2. $W^{\geq k}(MAJ) \leq \varepsilon$, $k = O(\frac{\sqrt{n}}{\varepsilon})$.

2.1 Decision Trees

A decision tree is a model of computation where the nodes are labeled with variables, leaf nodes are labeled +1 or −1; each node has out-degree 2, including an edge representing +1 and another representing −1.

There are two complexity measures that we care about, the *depth* which is the length of the longest root-to-leaf path, and the *size* which the number of edges.

For a function $f : \{-1, 1\}^n \to \{-1, 1\}$, define

$$DT_{depth}(f) = \min\{\text{depth of decision tree } T : T \text{ computes } f\}$$

and define

$$DT_{size}(f) = \min\{\text{size of decision tree } T : T \text{ computes } f\}$$

[1] Some of the results here also generalize to $\mathbb{R}$-valued functions, but for most applications, we focus on Boolean valued functions.

We also define the degree of a function. Let $f : \{-1,1\}^n \to \{-1,1\}$, the degree of f is the highest degree of monomials, i.e.

$$\deg(f) = \max_{S:\hat{f}(S)\neq 0} \{|S|\}$$

Claim 2.3. *Suppose f is computable by a depth d decision tree, then $W^{\geq d+1}[f] = 0$.*

The above claim can be rephrased as $\deg(f) \leq DT_{depth}(f)$. We will prove this claim in the next lecture, but before that, one can ask if the converse is true, that is, "is $DT_{depth}(f)$ upper bounded by the degree of the function in some form? This is still an open problem and the best bound we know so far is $DT_{depth}(f) \leq \deg(f)^4$.