Natural Proofs Barrier

CS640 Extra Talk

By: Sakaar Khurana

Instructor: Prof. Nitin Saxena

Topics Covered

- Basic introduction to Boolean Circuits
- Natural Proofs:
  - Why are circuit lower bounds so difficult? (Chapter 23, Computational Complexity: A Modern Approach)
  - Natural Proofs (Alexander A Razborov, Steven Rudich), 1994

Boolean Circuits and $P_{/poly}$

- **Boolean circuit:** An n input single output Boolean circuit is a directed acyclic graph where vertices are gates labelled with AND , OR or NOT. Size of a circuit denoted by $|C|$ is the number of vertices in it.
- **Circuit family:** A $T(n)$ size circuit family is a sequence $\{C_n\}_{n \in N}$ of Boolean circuits where C_n has n inputs and its size $C_n \leq T(n)$ for every n.
- **Language recognition:** We say that a language L is in $SIZE(T(n))$ if there exists a $T(n)$-size circuit family $\{C_n\}_{n \in N}$ such that for every $x \in \{0,1\}^n$,
  $x \in L \Leftrightarrow C_n(x) = 1.$
- $P_{/poly}$ (decidable by polynomial circuit families)= $\cup_c SIZE(n^c)$.

$P_{/poly}$, P and NP

- $P_{/poly}$ is decided by a Turing machine which takes advice in polynomial time.
- $P \subseteq P_{/poly}$
- $NP \nsubseteq P_{/poly}$ (conjectured) (if $NP \subseteq P_{/poly}$ then $PH = \Sigma_2^p$)
- Other circuit classes: NC, AC.

Circuit theory to solve P=NP - Motivation

- Why are problems like P=NP, P=PSPACE so difficult to solve?
  Known methods are inherently too weak to solve the problems such as P=NP.
- Baker, Gill, Solovay used oracle separation results for many major complexity classes to argue that relativizing proof techniques could not solve these problems.
- People then began to study these problems from the vantage of Boolean circuit complexity.
- New goal: A stronger non uniform version of P=NP, namely SAT does not have polynomial size circuits.
- Many proof techniques have been successfully applied to prove lower bounds in circuit complexity (all such known proofs are “natural”).
- These techniques are not subject to relativization.
- There for every n>1 exists function $f:\{0,1\}^n \rightarrow \{0,1\}$ that cannot be by a circuit of size $\frac{2^n}{10n}$.

A General approach to solve P=NP

- Formulate some mathematical notion of a “property” of Boolean functions.
- Show that polynomial sized circuits cannot compute Boolean functions with the above “property”
- Show that SAT or some other NP-Complete problem satisfies the above “property”

Formalizing:

- Let P be the property such that $P(f) = 1$ for a function f satisfying property P.
- The property P satisfies: $P(g) = 0 \; \forall \; g \in SIZE(n^c)$. (Such a property is called n^c-useful)
- Show that $P(SAT) = 1$.

This is the general framework that is used by any proof to prove some circuit lower bound.

We now define natural proofs.

Natural Proofs: definition

A NATURAL PROOF is a proof along the same lines (of previous slide) BUT with the property P satisfying following 2 conditions:

- **Constructiveness:** There is an $2^{O(n)}$ time algorithm that on input the truth table of a function $g: \{0,1\}^n \to \{0,1\}$ outputs $P(g)$. (Truth table has size 2^n so algorithm runs in time polynomial the input size.)
- **Largeness:** The probability that a random function $g: \{0,1\}^n \to \{0,1\}$ satisfies $P(g) = 1$ is at least $\frac{1}{n}$.

MAIN THEOREM

- ***If $2^{n^{\varepsilon}}$ hard one-way functions exist. Then there exists a constant $c \in N$ such that there is no n^{c}-useful property P.***

- So this proves that if the conjecture is true then there can be no natural proof for $NP \nsubseteq P_{/poly}$.

Definition 9.4 *(One way functions)*
A polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ is a *one-way function* if for every probabilistic polynomial-time algorithm A there is a negligible function $\epsilon : \mathbb{N} \to [0,1]$ such that for every n,

$$\Pr_{\substack{x \in_R \{0,1\}^n \\ y = f(x)}} [A(y) = x' \text{ s.t. } f(x') = y] < \epsilon(n)\,.$$

Conjecture 9.5
There exists a one-way function.

Complexity measure

- We formalize "complicatedness" of a Boolean function as a function μ that maps every Boolean function on $\{0,1\}^n$ to a non-negative integer.
- μ is a formal complexity measure if it satisfies:
  - $\mu(x_i) \leq 1$ and $\mu(\overline{x}_i) \leq 1$ (trivial functions)
  - $\mu(f\ AND\ g) \leq \mu(f) + \mu(g)\ \forall\ f, g$
  - $\mu(f\ OR\ g) \leq \mu(f) + \mu(g)\ \forall\ f, g$
- If μ is a formal complexity measure then $\mu(f)$ is a lower bound on the formula complexity of f.
- If $\mu(f) \geq S$ for some f, then for at least ¼ of all functions $g: \{0,1\}^n \to \{0,1\}$ we must have $\mu(g) \geq \frac{S}{4}$.

  Proof: $f = h\ XOR\ g$ where $h = f\ XOR\ g$, so $f = (\bar{h}\ AND\ g)\ OR\ (h\ AND\ \bar{g})$
- Generalization: If $\mu(f) \geq S$, then for all $\varepsilon > 0$, at least $1 - \varepsilon$ of all functions $g: \{0,1\}^n \to \{0,1\}$ we must have $\mu(g) \geq Omega(\frac{S}{\left(n+\log\left(\frac{1}{\varepsilon}\right)\right)^2})$.

Largeness and Constructiveness

- Whenever size of a function $f: \{0,1\}^n \to \{0,1\}$ is at least S, then that also implies size of at least half of functions from $\{0,1\}^n \to \{0,1\}$ is greater than $\frac{S}{2} - 10$. Hence, lower bound on complexity of one function implies lower bound on complexity of half of the functions. Hence it gives intuition that probability that any random function possesses property P is non negligible, and tells why P should satisfy largeness.
- Constructiveness: The intuition behind constructiveness is that the majority of properties of Boolean functions or n-vertex graphs are at worst exponential, and also we don't yet understand mathematics of P outside exponential time. So this notion tries to encapsulate as many properties within the notion of natural as possible that we are comfortable working with.

Proof of the main theorem

- ***If $2^{n^{\varepsilon}}$ hard one-way functions exist. Then there exists a constant $c \in N$ such that there is no n^{c}-useful property P.***
- Given a one way function that can't be inverted in $2^{n^{\varepsilon}}$, we can obtain a pseudo random function family $\{f_s\}_{s \in \{0,1\}^*}$ such that $f_s(.)\ for\ s \in_r \{0,1\}^m$ cannot be distinguished from a random function from $\{0,1\}^m \to \{0,1\}$ by $2^{m^{\varepsilon'}}$-time algorithm for some constant ε' with non-negligible probability. (Also, there is a polynomial time algorithm that given s, x outputs $f_s(x)$).
- Proof idea: Suppose P be a n^c useful natural property. We show that P can be used to distinguish between $f_s(.)\ for\ s \in_r \{0,1\}^m$ and a random function from $\{0,1\}^m \to \{0,1\}$ by $2^{m^{\varepsilon'}}$-time algorithm with non-negligible probability. Since $\mathbf{2^{n^{\varepsilon}}}$ hard one-way functions are conjectured to exist therefore P does not exist.

Proof of the main theorem

PROOF:

- Suppose P be a n^c useful natural property
- P can be thought of as an algorithm running in $2^{O(n)}$ time that
  - Outputs 0 on functions with circuit complexity lesser than n^c.
  - Outputs 1 on non-negligible number of functions.
- Let distinguisher has access to an oracle function h (which can either be a random function or a random function from the pseudo random family)
- Now distinguisher runs algorithm D as follows:
  - Let $n = m^{\epsilon/2}$ then construct truth table for g from $\{0,1\}^n \rightarrow \{0,1\}$ defined as: $g(x) = h(x0^{m-n})$.
  - D then runs P on this function g and outputs whatever P outputs.

Analyzing distinguisher

- If h was a random function then g is also a random function, therefore P and hence D outputs 1 with probability $\geq \frac{1}{n}$.
- If h was f_s for some s then function g has circuit complexity at most n^c since the map $s, x \rightarrow f_s(x)$ can be computed in $poly(m)$ time and hence map $x \rightarrow g(x)$ is computable by a circuit of size $poly(m) = n^c$. Hence D always outputs 0 in this case.
- Hence the distinguisher distinguishes with probability at least $\frac{1}{n}$ and takes $2^{O(n)} < 2^{m^{\varepsilon}}$ time.
- Hence natural property P cannot exist.
- Hence proved!

Unnatural proofs – intuition

- Subject to truth of hard pseudo-random generator conjecture:
  - Any proof that some function does not have small circuits must seize on some very specialized property i.e. one shared by negligible fraction of functions.

  OR

  - The proof must define a very complicated property, one that is outside the bounds of most mathematical experience(not exponential).
- So the proof must be unnatural by violating either largeness or constructivity.

Parameterized natural proof

- Let S and T be complexity classes. Then we call a combinatorial property T-natural if it is constructible in time T.
- Usefulness: For any Boolean function f such that $P(f) = 1$ then $f \notin S$.
- So, a lower bound proof that some explicit function is not in S is called T-natural if it states a T-natural property P and is useful against S.

Circuit lower bounds for other classes

- Following the same lines of the main proof before it is clear that any complexity class that has plausible pseudo-random function generator can't be used to prove circuit lower bounds.
- Hence in the parameterized framework defined before, we can have a natural proof only if class S does not have a plausible pseudo-random function generator.
- Proving circuit lower bounds comes stops at AC.
- Almost all circuit bounds follow from natural proofs or are naturalizable.

THANK YOU ☺