Lower bounds for monotone circuits (for CS 721- Computational Complexity)

Presentation · October 2018

Meet Taraviya
Indian Institute of Technology Bombay

Lower bounds for monotone circuits

Meet Taraviya and Mukesh Pareek

IIT Bombay

CS 721 - Computational Complexity

March 22, 2019

Overview

Monotone functions and circuits

Definition

For $x, y \in \{0,1\}^n$, we denote $x \preccurlyeq y$ if every bit that is 1 in x is also 1 in y. A function $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if $f(x) \leq f(y)$ for every $x \preccurlyeq y$.

Definition

A boolean circuit is said to be *monotone* if it only contains AND and OR gates.

Theorem

Every monotone circuit computes a monotone function, and every monotone function can be computed by a (sufficiently large) monotone circuit.

Motivation

- $NP \not\subset P_{/poly} \implies P \neq NP$
- If NP does not have polynomial-size circuits then $NP \not\subset P_{/poly}$
- The aim is to find problems in NP that are hard for poly-size circuits
- Best known lower bounds on non-uniform circuit size for problems in NP is linear, no super polynomial bounds known for even NEXP
- It is believed that the lower bound is exponential
- Razborov proved super polynomial monotone circuit bounds for the NP-complete problem CLIQUE (defined later) [Raz85]
- This was improved by Alon & Bopanna to show exponential bound for CLIQUE [AB87]

The CLIQUE problem

Clique in a graph

In the mathematical area of graph theory, a **clique** is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are adjacent; that is, its induced subgraph is complete.

The CLIQUE function

The *clique function* $f_n = CLIQUE(n, k)$ has $\binom{n}{k}$ variables x_{ij}, one for each potential edge in a graph on n vertices [n] $= \{1,...,n\}$; the function outputs 1 iff the associated graph contains a clique (complete subgraph) on some k vertices.

Monotonicity of CLIQUE

The clique function is monotone because setting more edges to 1 can only increase the size of the larges clique. If a graph has a clique of size k, the clique can't vanish on adding an edge.

The CLIQUE problem

Clique is NP-complete

The clique decision problem is NP-complete. It was one of Richard Karp's original 21 problems shown NP-complete.

Proof of NP-completeness

The proof shows a many-one reduction from the Boolean satisfiability problem, which was shown to be NP-complete by Cook-Levin.

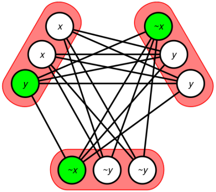

Figure: The 3-CNF satisfiability instance reduced to Clique. The green vertices form a 3-clique and correspond to a satisfying assignment.

Theorem 1

A weaker problem

Prove that clique decision problem is hard to compute for monotone circuits. Monotone circuits are weaker than general circuits. Originally considered with a hope to extend the results to general circuits.

Monotone-circuit lower bound for CLIQUE [Raz85a, And85, AB87]

Theorem : There exists some constant $\epsilon > 0$ such that for every $k \leq n^{1/4}$, there's no monotone circuit of size less than $n^{\sqrt{k}}$ that computes $CLIQUE_{n,k}$; i.e. exponential monotone circuit lower bound for clique.

Proof Terminology

Clique Indicators

For every $S \subseteq [n]$, C_S denotes the function on $\{0,1\}^{\binom{n}{2}}$ that outputs 1 on a graph G iff the set S is a clique in G and is called the clique indicator of S.
Note : $CLIQUE_{n,k} = \vee_{S \subseteq [n], |S|=k} C_S$

$\mathcal{Y}$: Distribution of Positive Graphs

It is the distribution of special graphs containing cliques on k vertices. Pick a set $K \subseteq [n]$ with $|K| = k$ at random. Output a graph that has a clique on vertices in K, and no other edges. $Pr[CLIQUE_{n,k}(\mathcal{Y}) = 1] = 1$

$\mathcal{N}$: Distribution of Negative Graphs

It is the distribution of special graphs with no clique of size k. Pick a function $c : [n] \rightarrow [k-1]$ at random. Output a graph that has an edge between u and v iff $c(u) \neq c(v)$. $Pr[CLIQUE_{n,k}(\mathcal{N}) = 0] = 1$

Circuit Approximator

To analyze the circuit, we approximate every small monotone circuit by a special type of monotone circuits characterized by DNFs.
Note : $C_S = \wedge_{i \neq j \in S} x_{ij}$; is a monomial depending on $\binom{|S|}{2}$ *variables*

(m,l)-*approximator*

An (m,l)-*approximator*, is an OR of at most m clique indicators, each of whose underlying vertex sets have cardinality at most l:

$$A = \vee_{t=1}^{r} C_{S_t} = \vee_{t=1}^{r} \wedge_{i \neq j \in S_t} x_{ij} \quad (r \leq m, |S_t| \leq l) \tag{1}$$

$l \geq 2$ and $m \geq 2$ are parameters depending only on values of k and n; which will be fixed later to complete the proof

Proof Outline

We start by assuming that there exists a monotone circuit F computing $f_n = CLIQUE(n, k)$, and let F' be the approximated circuit, that is, an (m,l)-*approximator* of the last gate of F. We show that:

- Every approximator (including F') must make a lot of errors, that is, disagree with f_n on many negative an positive graphs.
- If size(F) is small, then F' cannot make too many errors.

This will imply that size(F) must be large.

Lemma 1

Every approximator either rejects all graphs or wrongly accepts at least a fraction $1 - l^2/(k-1)$ of all $(k-1)^n$ negative graphs.

Proof : Let $A = \vee_{i=1}^{r} C_{S_i}$ be an (m,l)-*approximator*, and assume that A accepts at least one graph. Then $A \geq C_{S_1}$.
We have $\binom{|S_1|}{2}$ pairs of vertices in S_1 and for each such pair at most $(k-1)^{n-1}$ colorings assign the same color. Thus at most, $\binom{|S_1|}{2}(k-1)^{n-1} \leq \binom{l}{2}(k-1)^{n-1}$ negative graphs can be rejected by C_{S_1} and hence, by the approximator A.
Thus, every approximator (including F') must make a lot of errors.

Constructing the approximator F'

Given a monotone circuit F of size s for the $CLIQUE_{n,k}$, we will construct an (m, l) approximator F' for F in a "bottom-up" manner, starting from the input variables. Approximator for input variable x_{ij} will be $C_{\{i,j\}}$.

For an internal node $f \vee g$ (resp. $f \wedge g$) we describe the construction of an (m, l) approximator $f \sqcup g$ (resp. $f \sqcap g$) such that F' does not make too many errors, i.e.

Lemma 2

The number of positive graphs wrongly rejected by F' is at most $s \cdot m^2 \binom{n-l-1}{k-l-1}$.

Lemma 3

The number of negative graphs wrongly accepted by F' is at most $s \cdot m^2 l^{2p} (k-1)^{n-p}$.

We will also use **sunflower lemma** in our construction.

Sunflower lemma

Theorem

Let $\mathcal{Z}$ be a collection of distinct sets each of cardinality at most l. If $|\mathcal{Z}| > (p-1)^l l!$ then there exist p sets $Z_1, ..., Z_p \in \mathcal{Z}$ and set Z such that $Z_i \cap Z_j = Z$ for every $1 \leq i < j \leq p$.

Proof.

By induction. $l = 1 : Z = \phi$ works.

Assume the statement is true for $l = k - 1$. For $l = k$, assume we have $\mathcal{M} \subseteq \mathcal{Z}$, a maximal set of pairwise disjoint sets. If $|\mathcal{M}| \geq p$, we have $Z = \phi$. Otherwise, each $x \in \cup\mathcal{M}$ occurs in some $Z \in \mathcal{Z}$ (by maximality). $|\cup \mathcal{M}| \leq k(p-1)$. Hence some x occurs in more than $\frac{(p-1)^k k!}{(p-1)k} = (p-1)^{k-1}(k-1)!$ sets in $\mathcal{Z}$. After removing x from these sets, each set will be of size at most $k-1$ and hence have a sunflower of size p with $k-1$ elements in each petal. Adding x to each petal gives a sunflower with k elements in each petal. □

Lower bound for no sunflower

Theorem

There is a $\mathcal{Z}$, a collection of size $(p-1)^l$ of distinct sets each of cardinality at most l, with no sunflower with p petals.

Proof.

$\mathcal{Z} = \{\{(i, f(i)) | i \in [l]\} | f : [l] \rightarrow [p-1]\}$ Consider $\mathcal{M} \subseteq \mathcal{Z}$ where $|\mathcal{M}| = p$. Let (i, j) be an element not present in all sets in $\mathcal{M}$. There are $p-1$ elements of the form $(i, *)$. So there are $M, M' \in \mathcal{M}$ such that $(i, j) \in M \cap M'$ for some j, but $(i, j) \notin \cap \mathcal{M} \Rightarrow \mathcal{M}$ is not a sunflower. □

$f \sqcup g$

If f and g are (m, l)-functions, such that

$$f = \bigvee_{i=1}^{m} C_{S_i}, g = \bigvee_{j=1}^{m} C_{T_j}$$

$h = f \vee g$ has at most $2m$ clauses, and hence may not be a (m, l) function. So we repeatedly replace groups of clauses $C_{Z_1} \dots C_{Z_p}$ with a stronger clause C_Z using **sunflower lemma**, until the number of clauses left is at most m. We call this procedure *plucking*. We define $f \sqcup g$ as the function obtained after plucking. To be able to apply sunflower lemma, we set $m := l!(p-1)^l$.

$f \sqcap g$

If f and g are (m, l)-functions, such that

$$f = \bigvee_{i=1}^{m} C_{S_i}, g = \bigvee_{j=1}^{m} C_{T_j}$$

we define

$$h = \bigvee_{i=1}^{m} \bigvee_{j=1}^{m} C_{S_i \cup T_j}$$

which has at most m^2 clauses. We remove clauses C_Z with $|Z| > l$ and reduce the number of clauses to at most m by repeatedly applying the sunflower lemma as before (*plucking*). We define $f \sqcap g$ as the function obtained by this procedure. Note that

$$f \wedge g = \bigvee_{i=1}^{m} \bigvee_{j=1}^{m} C_{S_i} \wedge C_{T_j} \neq h$$

Lemma 2

We defined $f \sqcup g$ by replacing some clauses from $f \vee g$ with a weaker clause. So $f \sqcup g$ does not wrongly reject **positive** graphs. Thus plucking does not introduce false negatives.

To approximate $f \wedge g$, we first replace $C_{S_i} \wedge C_{T_j}$ with $C_{S_i \cup T_j}$, which behave identically on positive graphs. Hence this step does not introduce false negatives. Then we remove clauses with $|S_i \cup T_j| > l$. Because of this, we wrongly reject positive graphs in which $S_i \cup T_j$ is a clique - there are at most $\binom{n-l-1}{k-l-1}$ such graphs. Since we remove at most m^2 clauses, we wrongly reject at most $m^2\binom{n-l-1}{k-l-1}$ positive graphs. After this, we do plucking, which does not introduce any false negatives. Thus approximating $f \wedge g$ using $f \sqcap g$ introduces at most $m^2\binom{n-l-1}{k-l-1}$ false negatives.

Since there are at most s AND gates, F' wrongly rejects at most $s \cdot m^2\binom{n-l-1}{k-l-1}$ positive graphs.

Wrongly accepted negative graphs when approximating $f \vee g$ using $f \sqcup g$? We will show that each plucking $Z_1, ..., Z_p \to Z$ increases this number by at most $l^{2p}(k-1)^{n-p}$ and we will do at most $2m$ such pluckings in one approximation step $\Rightarrow$ at most $2ml^{2p}(k-1)^{n-p}$ wrongly accepted negative graphs OR gate.

Z must be a clique and none of $Z_i s$ is a clique. We defined $G \in \mathcal{N}$ using a random function $c : [n] \to [k-1]$ with an edge between u and v whenever $c(u) \neq c(v)$. So we need c to be one-to-one on Z (event B) without being one to one on any Z_i (event A_i). $Pr[A_i|B] =$ probability of collision in $Z_i \setminus Z \leq \frac{l^2}{k-1}$. Since $Z_i \setminus Z$ are disjoint,
$Pr[A_1 \wedge ... \wedge A_p \wedge B] \leq Pr[A_1 \wedge ... \wedge A_p | B] = \prod_{i=1}^{p} Pr[A_i|B] \leq l^{2p}(k-1)^{-p}$.

Lemma 3 (contd.)

For calculating the approximator $f \sqcap g$, replacing $C_{S_i} \wedge C_{T_j}$ with $C_{S_i \cup T_j}$ or removing clauses with $|S_i \cup T_j| > l$ does not introduce any false positives. Each plucking introduces at most $l^{2p}(k-1)^{n-p}$ false positives, with at most m^2 pluckings. Thus approximating AND gates introduces at most $m^2 l^{2p}(k-1)^{n-p}$ false positives on negative graphs.

Thus each gate introduces at most $m^2 l^{2p}(k-1)^{n-p}$ false positives on negative graphs. Hence F' wrongly accepts at most $s \cdot m^2 l^{2p}(k-1)^{n-p}$ negative graphs.

Main Theorem

Theorem

For $3 \leq k \leq n^{1/4}$, the monotone circuit complexity of $CLIQUE(n,k)$ is $n^{\Omega(\sqrt{k})}$

Proof.

Let F be a monotone circuit of size s deciding $CLIQUE(n,k)$. Construct F' as described using $l = \lfloor \frac{\sqrt{k-1}}{2} \rfloor$, $p = \Theta(\sqrt{k} \log n)$ and $m = l!(p-1)^l \leq (pl)^l$. By lemma 1, there are 2 cases.

If F' is identically 0, applying lemma 2 gives $s \cdot m^2 \binom{n-l-1}{k-l-1} \geq \binom{n}{k} \Rightarrow s$ is $n^{\Omega(\sqrt{k})}$. (Because $\binom{n}{k} / \binom{n-x}{k-x} \geq (n/k)^x$).

If F' outputs 1 on at least $(1 - \frac{l^2}{k-1} \geq \frac{1}{2})$ fraction of all negative graphs, applying lemma 4 gives $s \cdot m^2 2^{-p}(k-1)^n \geq \frac{1}{2}(k-1)^n \Rightarrow s$ is $n^{\Omega(\sqrt{k})}$. □

Very large size cliques are easy to detect

Theorem

For every constant k, the function CLIQUE(n,n-k) can be computed by a monotone formula containing at most $\mathcal{O}(n^2 logn)$ gates. The number of gates remains polynomial in n as long as k $= \mathcal{O}(\sqrt{logn})$; cliques of size $n - k$ are easy to detect when k is small. [Andreev-Jukna 2008]

Proof : We consider the dual of the function CLIQUE(n,n-k)
Dual of a boolean function $f(x_1, ..., x_n)$ is the function
$f^*(x_1, ...x_n) = \neg f(\neg x_1, ..., \neg x_n)$
Dual of CLIQUE(n,n-k) accepts a given graph G on n vertices iff G has no independent set with n-k vertices $\implies$
Vertex cover number of G: $\tau(G) \geq k + 1$
This problem can be solved by montonic formula of polynomial size.

Implications & Further Work

$NP \neq P$

$$(P \subseteq P/poly = PSIZE) \wedge (NP \not\subseteq PSIZE) \implies P \neq NP$$

$NP \not\subseteq BPP$

$$(BPP \subseteq P/poly) \wedge (NP \not\subseteq PSIZE) \implies NP \not\subseteq BPP$$

Open Problem

Whether this results holds for PSIZE; class of languages computable by polynomial size general circuits is still an open problem.

Questions?

Thank You!