Lecture 7
April 25, 2023

CS151
Complexity
Theory

1

Monotone circuits

- A question:

Do all
poly-time computable monotone functions
have
poly-size monotone circuits?

  – recall: true in non-monotone case

2

Monotone circuits

A monotone circuit for $\text{CLIQUE}_{n,k}$

- Input: graph G = (V,E) as adj. matrix, |V|=n
  – variable $x_{i,j}$ for each possible edge (i,j)
- ISCLIQUE(S) = monotone circuit that = 1 iff S ⊆ V is a clique: $\bigwedge_{i,j \in S} x_{i,j}$

$\text{CLIQUE}_{n,k}$ computed by monotone circuit:

$\bigvee_{S \subseteq V, |S|=k}$ ISCLIQUE(S)

3

Monotone circuits

- Theorem (Razborov 85): monotone circuits for $\text{CLIQUE}_{n,k}$ with k = $n^{1/4}$ must have size at least

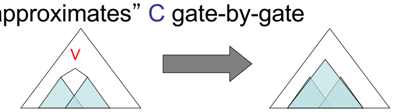
$2^{\Omega(n^{1/8})}$.

- Proof:
  – rest of lecture

4

Proof idea

- "method of approximation"
- suppose C is a monotone circuit for $\text{CLIQUE}_{n,k}$
- build another monotone circuit CC that "approximates" C gate-by-gate



5

Proof idea

- on test collection of positive/negative instances of $\text{CLIQUE}_{n,k}$:
  – local property: few errors at each gate
  – global property: many errors on test collection

- Conclude: C has many gates

6

Notation

- input: graph G = (V, E)
- variable $x_{j,k}$ for each potential edge (j, k)
- CC($X_1, X_2, \ldots X_m$), where $X_i \subseteq V$, means:

$$\bigvee_i \left(\bigwedge_{j,k \in X_i} x_{j,k}\right)^{*}$$

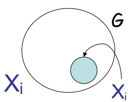
- For example: CC($X_1, X_2, \ldots X_m$) where the X_i range over all k-subsets of V
  - this is the obvious monotone circuit for $\text{CLIQUE}_{n,k}$ from a previous slide.

*[CC( ) = 0; $(\bigwedge_{i,j \in \emptyset} x_{i,j})$= 1]

7

Preview

- approximate circuit CC($X_1, X_2, \ldots X_m$)
- n = # nodes
- k = $n^{1/4}$ = size of clique
- h = $n^{1/8}$ = max. size of subsets X_i
  - this is "global property" that ensures lots of errors
  - many graphs G with no k-cliques, but clique on X_i of size h



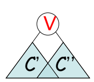
8

Preview

- approximate circuit CC($X_1, X_2, \ldots X_m$)
- p = $n^{1/8}$log n
- M = $(p – 1)^h h!$
- max # of subsets is M (so m ≤ M)
  - critical for "local property" that ensures few errors at each gate

9

Approximate OR



CC($X_1, X_2, \ldots X_{m'}$)   CC($Y_1, Y_2, \ldots Y_{m''}$)

- exact OR:

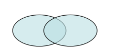
CC($X_1, X_2, \ldots X_{m'}, Y_1, Y_2, \ldots Y_{m''}$)

  - set sizes still ≤ h
  - may be up to 2M sets; need to reduce to M

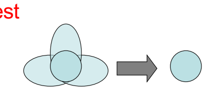
10

Approximate OR

- throw away sets? bad:many errors
- throw away overlapping sets? – better



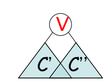
- throw away special configuration of overlapping sets – best



11

Approximate OR

- CC($X_1, X_2, \ldots X_{m'}$)
- CC($Y_1, Y_2, \ldots Y_{m''}$)



- exact OR:

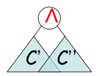
CC($X_1, X_2, \ldots X_{m'}, Y_1, Y_2, \ldots Y_{m''}$)

  - while more than M sets, find (h, p)-sunflower; replace with its core ("**pluck**")
- approximate OR:

CC(**pluck**($X_1, X_2, \ldots X_{m'}, Y_1, Y_2, \ldots Y_{m''}$))

12

Approximate AND



- CC($X_1, X_2, \ldots X_{m'}$)
- CC($Y_1, Y_2, \ldots Y_{m''}$)
- (close to) exact AND:

  CC($\{(X_i \cup Y_j) : 1 \le i \le m', 1 \le j \le m''\}$)

  – some sets may be larger than h; discard them

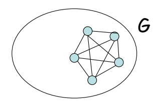
  – may be up to M^2 sets. While > M sets, find (h, p)-sunflower; replace with its core ("**pluck**")
- approximate AND:

  CC(**pluck** ($\{(X_i \cup Y_j) : |X_i \cup Y_j| \le h\}$))

13

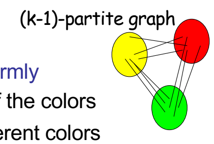
Test collection

- Positive instances: all graphs G on n nodes with a k-clique and no other edges.



14

Test collection



- Negative instances:
  – k-1 colors
  – color each node uniformly at random with one of the colors
  – edge (x, y) iff x, y different colors
  – no k-clique
  – include graphs in their multiplicities
    - makes analysis easier

15

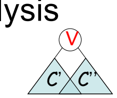
"Local" analysis

- "false positive":
  – negative example
  – gate is supposed to output 0, but our CC outputs 1

<u>Lemma</u>: each approximation step introduces at most $M^2(k-1)^n/2^p$ false positives.

16

"Local" analysis

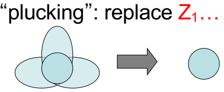


- Proof:
  – case 1: OR

  CC($X_1, X_2, \ldots X_{m'}$) CC($Y_1, Y_2, \ldots Y_{m''}$)

  CC(**pluck**($X_1, X_2, \ldots X_{m'}, Y_1, Y_2, \ldots Y_{m''}$))

  – given "plucking": replace $Z_1 \ldots Z_p$ with Z



  – bad case: clique on Z, and each petal is missing at least one edge

17

"Local" analysis

– what is the probability of a repeated color in each Z_i but no repeated colors in Z?

event R(S) = repeated colors in S

$\Pr[R(Z_1) \wedge R(Z_2) \ldots R(Z_p) \wedge \neg R(Z)]$

$\le \Pr[R(Z_1) \wedge R(Z_2) \ldots R(Z_p) | \neg R(Z)]$

(definition of conditional probability)

$= \prod_i \Pr[R(Z_i) \mid \neg R(Z)]$

(independent events given no repeats in Z)

$\le \prod_i \Pr[R(Z_i)]$

(obviously larger)

18

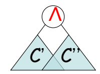
“Local” analysis

– for every pair of vertices in Z_i, probability of same color is $1/(k-1)$
– $R(Z_i) \le (h \text{ choose } 2)/(k-1) \le ½$
– $\prod_i \Pr[R(Z_i)] \le (½)^p$
– # negative examples is $(k-1)^n$
– # false positives in given plucking step is at most $(½)^p(k-1)^n$
– at most M plucking steps
– # false positives at OR ≤ $M(½)^p(k-1)^n$

19

“Local” analysis

– case 2: AND



$CC(X_1,X_2,\ldots X_{m'})$ $CC(Y_1,Y_2,\ldots Y_{m''})$

$CC(\mathbf{pluck}(\{(X_i \cup Y_j) : |X_i \cup Y_j| \le h\}))$

– discarding sets $(X_i \cup Y_j)$ larger than h can only make circuit accept fewer examples
  - no false positives here

20

“Local” analysis

– up to M^2 pluckings
– each introduces at most $(½)^p(k-1)^n$ false positives (previous slides)

– # false positives at AND ≤ $M^2(½)^p(k-1)^n$

21

“Local” analysis

- “false negative”:
  – positive example;
  – gate is supposed to output 1, but our CC outputs 0

Lemma: each approximation step introduces at most

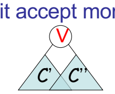
$$M^2\binom{n-h-1}{k-h-1}$$

false negatives.

22

“Local” analysis

- Proof:
  – Case 1: OR
  – plucking can only make circuit accept more examples
    - no false negatives here.



  – Case 2: AND

$CC(X_1,X_2,\ldots X_{m'})$ $CC(Y_1,Y_2,\ldots Y_{m''})$

$CC(\mathbf{pluck}(\{(X_i \cup Y_j) : |X_i \cup Y_j| \le h\}))$

  - for positive examples: clique on X_i and clique on Y_j ⇒ clique on $X_i \cup Y_j$ (no false negatives until discard $X_i \cup Y_j$ sets)

23

“Local” analysis

– discarding set $Z = (X_i \cup Y_j)$ larger than h may introduce false negatives
– any clique that includes Z is a problem; there are at most

$$\binom{n-|Z|}{k-|Z|} \le \binom{n-h-1}{k-h-1}$$

such positive examples, since $|Z|>h$ & $h \ll k$
– at most M^2 such deletions
– we’ve seen plucking doesn’t matter

24

"Global" analysis

Lemma: every non-trivial CC outputs 1 on at least ½ of the negative examples.

- Proof:
  - CC contains some set X of size at most h
  - accepts all neg. examples with different colors in X
  - probability X has repeated colors is

    R(X) ≤ (h choose 2)/(k-1) ≤ ½
  - probability over negative examples that CC accepts is at least ½.

25

Finishing up

- First possibility: trivial CC, rejects all positive examples
  - every positive example must have been false negative at some gate
  - number of gates must be at least:

$$\underbrace{\binom{n}{k}}_{\text{\# of positive examples}} / M^2 \underbrace{\binom{n-h-1}{k-h-1}}_{\text{false negatives at each gate}}$$

26

Finishing up

- Second possibility: CC accepts at least ½ of negative examples
  - every negative example must have been false positive at some gate
  - number of gates must be at least:

$$\underbrace{\tfrac{1}{2}(k-1)^n}_{\text{\# of negative examples}} / M^2 \underbrace{2^{-p}(k-1)^n}_{\text{false positives at each gate}}$$

27

Finishing up

$$\binom{n}{k} / M^2 \binom{n-h-1}{k-h-1}$$

$$\frac{1}{2}(k-1)^n / M^2 2^{-p}(k-1)^n$$

Both quantities are at least $2^{\Omega(n^{1/8})}$

28

Conclusions

- A question (true in non-monotone case):

  Do all poly-time computable monotone functions have poly-size monotone circuits?
- if yes, then we would have just proved **P ≠ NP**
  - why?

29

Conclusions

- unfortunately, answer is no
- Razborov later showed similar (super-polynomial) lower bound for MATCHING, which is in **P…**

30

31

Randomness

- 3 examples of the power of randomness
  - communication complexity
  - polynomial identity testing
  - complexity of finding unique solutions
- randomized complexity classes

32

1. Communication complexity

two parties: Alice and Bob
function $f:\{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$
Alice holds $x \in \{0,1\}^n$; Bob holds $y \in \{0,1\}^n$

- Goal: compute f(x, y) while communicating as few bits as possible between Alice and Bob
- count number of bits exchanged (computation free)
- at each step: one party sends bits that are a function of held input and received bits so far

33

Communication complexity

- simple function (equality):

  EQ(x, y) = 1 iff x = y

- simple protocol:
  - Alice sends x to Bob (n bits)
  - Bob sends EQ(x, y) to Alice (1 bit)
  - total: n + 1 bits
  - (works for any predicate f)

34

Communication complexity

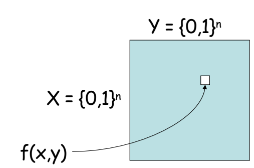
- Can we do better?
  - deterministic protocol?
  - probabilistic protocol?
    - at each step: one party sends bits that are a function of held input and received bits so far and the result of some coin tosses
    - required to output f(x, y) with high probability over all coin tosses

35

Communication complexity

Theorem: no deterministic protocol can compute EQ(x, y) while exchanging fewer than n+1 bits.

- Proof:
  - "input matrix":

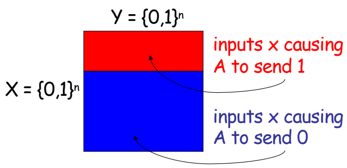


36

Communication complexity

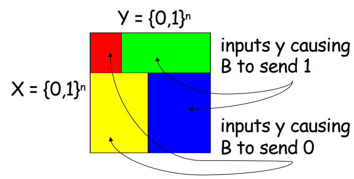
– assume 1 bit sent at a time, alternating (same proof works in general setting)

– A sends 1 bit depending only on x:



Communication complexity

– B sends 1 bit depending only on y and received bit:



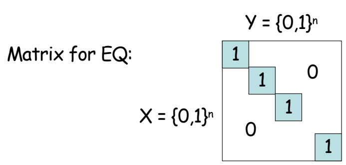
Communication complexity

– at end of protocol involving k bits of communication, matrix is partitioned into at most 2^k combinatorial rectangles

– bits sent in protocol are the same for every input (x, y) in given rectangle

– conclude: f(x,y) must be constant on each rectangle

Communication complexity



– any partition into combinatorial rectangles with constant f(x,y) must have $2^n + 1$ rectangles

– protocol that exchanges ≤ n bits can only create 2^n rectangles, so must exchange at least n+1 bits.

Communication complexity

- protocol for EQ employing randomness?
  – Alice picks random prime p in $\{1...4n^2\}$, sends:
    - p
    - (x mod p)
  – Bob sends:
    - (y mod p)
  – players output 1 if and only if:

    (x mod p) = (y mod p)

Communication complexity

– O(log n) bits exchanged

– if x = y, always correct

– if x ≠ y, incorrect if and only if:

  p divides |x – y|

– # primes in range is ≥ 2n

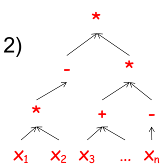
– # primes dividing |x – y| is ≤ n

– probability incorrect ≤ 1/2

Randomness gives an exponential advantage!!

2. Polynomial identity testing

- Given: polynomial $p(x_1, x_2, \ldots, x_n)$ as arithmetic formula (fan-out 1):
  - multiplication (fan-in 2)
  - addition (fan-in 2)
  - negation (fan-in 1)



Polynomial identity testing

- Question: Is p identically zero?
  - i.e., is $p(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathbf{F}^n$
  - (assume $|\mathbf{F}|$ larger than degree…)
- "polynomial identity testing" because given two polynomials p, q, we can check the identity $p \equiv q$ by checking if $(p - q) \equiv 0$

Polynomial identity testing

- try all $|\mathbf{F}|^n$ inputs?
  - may be exponentially many
- multiply out symbolically, check that all coefficients are zero?
  - may be exponentially many coefficients
- can randomness help?
  - i.e., flip coins, allow small probability of wrong answer

Polynomial identity testing

Lemma (Schwartz-Zippel): Let

$$p(x_1, x_2, \ldots, x_n)$$

be a total degree d polynomial over a field **F** and let S be any subset of **F**. Then if p is not identically 0,

$$\Pr_{r_1, r_2, \ldots, r_n \in S}[\, p(r_1, r_2, \ldots, r_n) = 0] \le d/|S|.$$

Polynomial identity testing

- Proof:
  - induction on number of variables n
  - base case: n = 1, p is univariate polynomial of degree at most d
  - at most d roots, so

$$\Pr[\, p(r_1) = 0] \le d/|S|$$

Polynomial identity testing

- write $p(x_1, x_2, \ldots, x_n)$ as

$$p(x_1, x_2, \ldots, x_n) = \Sigma_i (x_1)^i\, p_i(x_2, \ldots, x_n)$$

- k = max. i for which $p_i(x_2, \ldots, x_n)$ not id. zero
- by induction hypothesis:

$$\Pr[\, p_k(r_2, \ldots, r_n) = 0] \le (d-k)/|S|$$

- whenever $p_k(r_2, \ldots, r_n) \ne 0$, $p(x_1, r_2, \ldots, r_n)$ is a univariate polynomial of degree k

$$\Pr[p(r_1,r_2,\ldots,r_n)=0 \mid p_k(r_2,\ldots,r_n) \ne 0] \le k/|S|$$

Polynomial identity testing

$\Pr[\, p_k(r_2, \ldots, r_n) = 0] \le (d-k)/|S|$

$\Pr[p(r_1,r_2,\ldots,r_n)=0 \mid p_k(r_2,\ldots,r_n) \neq 0] \le k/|S|$

– conclude:

$\Pr[\, p(r_1, \ldots, r_n) = 0] \le (d-k)/|S| + k/|S| = d/|S|$

– Note: can add these probabilities because

$\Pr[E_1] = \Pr[E_1|E_2]\Pr[E_2] + \Pr[E_1|\neg E_2]\Pr[\neg E_2]$
$\le \Pr[E_2] + \Pr[E_1|\neg E_2]$

49