Outline

- CLIQUE
- monotone circuits and problems
- Razborov's lower bound for monotone circuits computing CLIQUE

Clique

Recall...
- \( IS = \{ (G, k) \mid G \text{ is a graph with an ind. set } V' \subseteq V \text{ of size } \geq k \} \)
  (independent set = set of vertices no 2 of which are connected by an edge)
- \( IS \) is \( \text{NP} \)-complete.
- \( CLIQUE = \{ (G, k) \mid G \text{ is a graph with a clique of size } \geq k \} \)
  (clique = set of vertices every pair of which are connected by an edge)
- \( CLIQUE \) is \( \text{NP} \)-complete.
  - reduction?

Circuit lower bounds

- We think that \( \text{NP} \) requires exponential-size circuits.
- Where should we look for a problem to attempt to prove this?
- Intuition: "hardest problems" - i.e., \( \text{NP} \)-complete problems
- Formally:
  - if any problem in \( \text{NP} \) requires super-polynomial size circuits
  - then every \( \text{NP} \)-complete problem requires super-polynomial size
  - poly-time reductions can be performed by poly-size circuits using a variant of \text{CVAL} construction

Monotone problems

- monotone language: language \( L \subseteq \{0,1\}^* \) such that \( x \in L \) implies \( x' \in L \) for all \( x \preceq x' \)
  - flipping a bit of the input from 0 to 1 can only change the output from "no" to "yes" (or not at all)
- some \( \text{NP} \)-complete languages are monotone
  - e.g. \text{CLIQUE} (given as adj. matrix):
  - others: \text{HAM. CYCLE, SET COVER...}
  - but not \text{SAT, KNAPSACK...}
### Monotone circuits

A restricted class of circuits:

- **monotone circuit**: circuit whose gates are ANDs ($\land$), ORs ($\lor$), but no NOTs
- can only compute monotone functions
  - monotone functions closed under AND, OR
- An interesting question: do all poly-time computable monotone functions have poly-size monotone circuits?
  - recall: true in non-monotone case

### Monotone circuits

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

- Input: graph $G = (V, E)$ as adjacency matrix, $|V| = n$
- variable $x_{i,j}$ for each possible edge $(i,j)$
- $\text{ISCLIQUE}(S) = \text{monotone circuit}$ that = 1 iff $S \subseteq V$ is a clique:
  $$\land_{i,j \in S} x_{i,j}$$
- $\text{CLIQUE}_{n,k}$ computed by monotone circuit:
  $$\lor_{S \subseteq V, |S| = k} \text{ISCLIQUE}(S)$$

### Monotone circuits

- Size of this monotone circuit for $\text{CLIQUE}_{n,k}$:
  $$\begin{pmatrix} n \\ k \end{pmatrix} \begin{pmatrix} k \\ 2 \end{pmatrix}$$
- when $k = n^{1/4}$, size is approx.:
  $$\left( \frac{n}{n^{1/4}} \right)^{n^{1/4}} \left( \frac{n^{1/4}}{2} \right)^2 \approx n^{\Omega(n^{1/4})}$$
- Theorem (Razborov): 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.

### Proof idea

- "method of approximation"
- suppose $C$ is a monotone circuit for $\text{CLIQUE}_{n,k}$
- build another monotone circuit $CC$ that "approximates" $C$
- on test collection of pos/neg 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
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:
  \[
  \text{OR}_i \text{ AND } \bigwedge_{j,k \in X_i} x_{j,k}
  \]
- e.g., $CC(X_1, X_2, \ldots X_m)$ where the $X_i$ range over all $k$-subsets of $V$:
  - the obvious monotone circuit for $\text{CLIQUE}_{n,k}$ from a previous slide.

Building $CC$

- $CC$ ("crude circuit") for circuit $C$ defined inductively as follows:
  - $CC$ for single variable $x$ is just $CC((x))$
  - no errors yet!
  - $CC$ for circuit $C$ of form:
    \[
    \begin{array}{c}
    C \cap C
    \end{array}
    \]
    - "approximate OR" of $CC$ for $C'$, $CC$ for $C$
  - $CC$ for circuit $C$ of form:
    \[
    \begin{array}{c}
    C \cup C
    \end{array}
    \]
    - "approximate AND" of $CC$ for $C'$, $CC$ for $C$
  - last 2 steps introduce errors

Approximate OR

- $CC(X_1, X_2, \ldots X_m)$:
  - exact OR:
    \[
    CC(X_1, X_2, \ldots X_m, Y_1, Y_2, \ldots Y_m)
    \]
    - set sizes still $\leq h$
    - may be up to 2M sets, need to reduce to $M$
    - throw away sets? bad - many errors
    - throw away overlapping sets - better
    - throw away special configuration of overlapping sets - best

Preview

- approx. 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 $h$
- $p = n^{1/8} \log n$
- $M = (p - 1)!h! = $ max $\#$ of subsets (so $m \leq M$)
  - critical for "local property" that ensures few errors at each gate
**Sunflowers**

- Definition: \((h, p)\)-sunflower is a family of \(p\) sets ("petals") each of size at most \(h\), such that intersection of every pair is a subset \(S\) (the "core").

- Lemma (Erdős-Rado): Every family of more than \(M = (p-1)^h h!\) sets, each of size at most \(h\), contains an \((h, p)\)-sunflower.

- Proof: not hard, in Papadimitriou.

**Approximate OR**

\[
CC(X_1, X_2, \ldots, X_m) \quad CC(Y_1, Y_2, \ldots, Y_{m'})
\]

- exact OR:
  \[
  CC(X_1, X_2, \ldots, X_m, Y_1, Y_2, \ldots, Y_{m'})
  \]
  - while \(M\) sets, find \((h, p)\)-sunflower; replace with its core ("pluck")

- approximate OR:
  \[
  CC(\text{pluck}(X_1, X_2, \ldots, X_m, Y_1, Y_2, \ldots, Y_{m'}))
  \]

**Approximate AND**

\[
CC(X_1, X_2, \ldots, X_m) \quad CC(Y_1, Y_2, \ldots, Y_{m'})
\]

- exact AND:
  \[
  CC(\{(X_i \cup Y_j) : 1 \leq i \leq m', 1 \leq j \leq m''\})
  \]
  - some sets may be larger than \(h\)
  - discard sets larger than \(h\)
  - may be more than \(M\) sets (up to \(M^2\))
  - while \(M\) sets, find \((h, p)\)-sunflower; replace with its core ("pluck")

- approximate AND:
  \[
  CC(\text{pluck}(\{(X_i \cup Y_j) : |X_i \cup Y_j| \leq h\}))
  \]

**Test collection**

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

- 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)
Analysis

• “false positive”: negative example; gate is supposed to output 0, but our $CC$ outputs 1
• Lemma: each approximation step introduces at most $M^2(k-1)^n/2^n$ false positives.
• Proof:
  - case 1: OR
    \[ CC(X_1, X_2, \ldots, X_m) \leq CC(Y_1, Y_2, \ldots, Y_m) \]
    \[ CC(\text{pluck}(X_1, X_2, \ldots, X_m, Y_1, Y_2, \ldots, Y_m)) \]
  - given "plucking": replace $Z_1 \ldots Z_p$ w/ $Z$
  - bad case: clique on $Z$, each petal missing at least one edge

Analysis

- given "plucking": replace $Z_1 \ldots Z_p$ w/ $Z$
- probability repeated color in each $Z_i$; no repeated colors in $Z$
- event $R(S)$ = repeated colors in $S$
\[ Pr[R(Z_1) \land R(Z_2) \land \ldots \land R(Z_p) \land \neg R(Z)] \]
\[ \leq Pr[R(Z_1) \land R(Z_2) \land \ldots \land R(Z_p) \land \neg R(Z)] \]
\[ \leq \prod_i Pr[R(Z_i) \mid \neg R(Z)] \] (defn of conditional probability)
\[ = \prod_i Pr[R(Z_i)] \] (independent events given no repeats in $Z$)
\[ \leq \prod_i Pr[R(Z_i)] \] (obviously larger)

Analysis

- given "plucking": replace $Z_1 \ldots Z_p$ w/ $Z$
- trying to bound $\prod_i Pr[R(Z_i)]$
- for every pair of vertices in $Z_i$, probability of same color = $1/(k-1)$
- $R(Z_i) \leq (h \choose 2)/(k-1) \leq 1/2$
- therefore $\prod_i Pr[R(Z_i)] \leq (1/2)^n$
- # neg. examples = $(k-1)^n$
- # false positives in given plucking step $\leq (1/2)^n(k-1)^n$
- at most $M$ plucking steps
- # false positives at OR $\leq M(1/2)^n(k-1)^n$

Analysis

• case 2: AND
\[ CC(X_1, X_2, \ldots, X_m) \leq CC(Y_1, Y_2, \ldots, Y_m) \]
\[ CC(\text{pluck}(\{X \cup Y \mid |X \cup Y| \leq h\})) \]
- discarding sets $(X \cup Y)$ larger than $h$
can only make circuit accept fewer examples - no false positives here
- up to $M^2$ pluckings
- each introduce at most $(1/2)^n(k-1)^n$ false positives (previous slides)
- # false pos.'s at AND $\leq M^2(1/2)^n(k-1)^n$
**Analysis**

- “false negative”: positive example; gate is supposed to output 1, but our CC outputs 0.
- Lemma: each approximation step introduces at most
  \[ M^2 \left( \frac{n-h-1}{k-h-1} \right) \]
  false negatives.
- Proof:
  - Case 1: OR
  - plucking can only make circuit accept more examples, no false negatives here.

**Analysis**

- case 2: AND
  \[ CC(X_1, X_2, \ldots, X_n) \quad CC(Y_1, Y_2, \ldots, Y_m) \]
  \[ CC(\text{pluck}( \{ X_i \cup Y_j : |X_i \cup Y_j| \leq h \} )) \]
  - 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
    \[ \binom{n-|Z|}{k-|Z|} = \binom{n-h-1}{k-h-1} \]
    such positive examples, since \( |Z| > h \)
  - at most \( M^2 \) such deletions; we’ve seen plucking doesn’t matter.

**Analysis**

- Lemma: every non-trivial CC outputs 1 on at least \( \frac{1}{3} \) of the negative examples.
- Proof:
  - CC contains some set \( X \) of size at most \( h \)
  - accepts all negative examples with different colors in \( X \)
  - from before: probability of two nodes in \( X \) same color = \( 1/(k-1) \)
  - probability \( X \) has repeated colors = \( R(X) \leq \binom{h}{2}/(k-1) \leq \frac{1}{2} \)
  - probability over negative examples that CC accepts is at least \( \frac{1}{2} \).

**Analysis**

- 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:
    \[ \binom{n}{k}/M^2 \binom{n-h-1}{k-h-1} \]
- Second possibility: CC accepts at least \( \frac{1}{3} \) of negative examples
  - every negative example must have been false positive at some gate
  - number of gates must be at least:
    \[ \frac{1}{2}(k-1)^n/M^2 2^{-p}(k-1)^n \]
  - Both quantities at least \( 2^{O(n^{1/3})} \).
Conclusions

- An interesting question: do all poly-time computable monotone functions have poly-size monotone circuits?
  - recall: true in non-monotone case

- if yes, then we would have just proved $P \neq NP$ (why?)

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