Lecture 14 - Counting I
Promise Problems, Unique-SAT

Rafael Oliveira
rafael.oliveira.teaching@gmail.com
University of Waterloo

CS 860 - Graduate Complexity Theory
Fall 2022

Overview

- Unique-SAT (Valiant-Vazirani)

- Unique-SAT (Valiant-Vazirani)

Unique-SAT

- Let $\mathcal{U}$ be the set of all satisfiable CNFs which have exactly one satisfying assignment

Unique-SAT

- Let $\mathcal{U}$ be the set of all satisfiable CNFs which have exactly one satisfying assignment
- Today: if we had a poly-time algorithm to solve all instances in $\mathcal{U}$, then we will show that $\text{NP} = \text{RP}$.

Unique-SAT (Valiant-Vazirani reduction)

- Let $\mathcal{U}$ be the set of all satisfiable CNFs which have exactly one satisfying assignment
- Today: if we had a poly-time algorithm to solve all instances in $\mathcal{U}$, then we will show that NP $=$ RP.
- randomized reduction from SAT to $\mathcal{U}$
  - from CNF ϕ we will construct polynomially many CNFs $\mathcal{F} := \{\varphi_0, \ldots, \varphi_m\}$

Unique-SAT (Valiant-Vazirani reduction)

- Let $\mathcal{U}$ be the set of all satisfiable CNFs which have exactly one satisfying assignment
- Today: if we had a poly-time algorithm to solve all instances in $\mathcal{U}$, then we will show that NP = RP.
- randomized reduction from SAT to $\mathcal{U}$
  - from CNF ϕ we will construct polynomially many CNFs $\mathcal{F} := \{\varphi_0, \ldots, \varphi_m\}$
  - If ϕ is satisfiable, with high probability $\mathcal{F} \cap \mathcal{U} \neq \emptyset$
  - If ϕ not satisfiable, then $\mathcal{F} \cap \mathcal{U} = \emptyset$ (always)

Unique-SAT

- Let $\mathcal{U}$ be the set of all satisfiable CNFs which have exactly one satisfying assignment
- Today: if we had a poly-time algorithm to solve all instances in $\mathcal{U}$, then we will show that $\mathrm{NP} = \mathrm{RP}$.
- randomized reduction from SAT to $\mathcal{U}$
  - from CNF ϕ we will construct polynomially many CNFs $\mathcal{F} := \{\varphi_0, \ldots, \varphi_m\}$
  - If ϕ is satisfiable, with high probability $\mathcal{F} \cap \mathcal{U} \neq \emptyset$
  - If ϕ not satisfiable, then $\mathcal{F} \cap \mathcal{U} = \emptyset$ (always)
- Main idea:
  1. $\phi \in$ SAT with n variables with $\sim 2^k$ satisfying assignments

Unique-SAT

- Let $\mathcal{U}$ be the set of all satisfiable CNFs which have exactly one satisfying assignment
- Today: if we had a poly-time algorithm to solve all instances in $\mathcal{U}$, then we will show that NP = RP.
- randomized reduction from SAT to $\mathcal{U}$
  - from CNF ϕ we will construct polynomially many CNFs $\mathcal{F} := \{\varphi_0, \ldots, \varphi_m\}$
  - If ϕ is satisfiable, with high probability $\mathcal{F} \cap \mathcal{U} \neq \emptyset$
  - If ϕ not satisfiable, then $\mathcal{F} \cap \mathcal{U} = \emptyset$ (always)
- Main idea:
  1. $\phi \in$ SAT with n variables with $\sim 2^k$ satisfying assignments
  2. let $h : \{0,1\}^n \to \{0,1\}^k$ be hash function picked from pairwise independent family

Unique-SAT

- Let $\mathcal{U}$ be the set of all satisfiable CNFs which have exactly one satisfying assignment
- Today: if we had a poly-time algorithm to solve all instances in $\mathcal{U}$, then we will show that NP = RP.
- randomized reduction from SAT to $\mathcal{U}$
  - from CNF ϕ we will construct polynomially many CNFs $\mathcal{F} := \{\varphi_0, \ldots, \varphi_m\}$
  - If ϕ is satisfiable, with high probability $\mathcal{F} \cap \mathcal{U} \neq \emptyset$
  - If ϕ not satisfiable, then $\mathcal{F} \cap \mathcal{U} = \emptyset$ (always)
- Main idea:
  1. $\phi \in$ SAT with n variables with $\sim 2^k$ satisfying assignments
  2. let $h : \{0,1\}^n \to \{0,1\}^k$ be hash function picked from pairwise independent family
  3. Expect exactly one assignment $x \in \{0,1\}^n$ such that

$$\phi(x) = 1 \ \text{ and } \ h(x) = 0$$

Unique-SAT

- Let $\mathcal{U}$ be the set of all satisfiable CNFs which have exactly one satisfying assignment
- Today: if we had a poly-time algorithm to solve all instances in $\mathcal{U}$, then we will show that NP = RP.
- randomized reduction from SAT to $\mathcal{U}$
  - from CNF ϕ we will construct polynomially many CNFs $\mathcal{F} := \{\varphi_0, \ldots, \varphi_m\}$
  - If ϕ is satisfiable, with high probability $\mathcal{F} \cap \mathcal{U} \neq \emptyset$
  - If ϕ not satisfiable, then $\mathcal{F} \cap \mathcal{U} = \emptyset$ (always)
- Main idea:
  1. $\phi \in$ SAT with n variables with $\sim 2^k$ satisfying assignments
  2. let $h : \{0,1\}^n \to \{0,1\}^k$ be hash function picked from pairwise independent family
  3. Expect exactly one assignment $x \in \{0,1\}^n$ such that

     $$\phi(x) = 1 \text{ and } h(x) = 0$$

  4. Construct CNF ψ from ϕ and h which is satisfied precisely by the assignment above

Pairwise Independent Hash Family

Definition 1 (Pairwise Independent Hash Family)

A family $\mathcal{H}$ of functions $h : \{0,1\}^n \to \{0,1\}^m$ is a pairwise independent family of hash functions if for every two different inputs $x, y \in \{0,1\}^n$ and every $a, b \in \{0,1\}^m$, we have

$$\Pr_{h \in \mathcal{H}}[h(x) = a \wedge h(y) = b] = \frac{1}{2^{2m}}$$

- When we pick h at random, the random variables $h(x)$ and $h(y)$ are independent and uniformly distributed.

Pairwise Independent Hash Family

Definition 1 (Pairwise Independent Hash Family)

A family $\mathcal{H}$ of functions $h : \{0,1\}^n \to \{0,1\}^m$ is a pairwise independent family of hash functions if for every two different inputs $x, y \in \{0,1\}^n$ and every $a, b \in \{0,1\}^m$, we have

$$\Pr_{h \in \mathcal{H}}[h(x) = a \wedge h(y) = b] = \frac{1}{2^{2m}}$$

Example 2

The family

$$\mathcal{H} := \{h_{a,b}(x) = (a_1 \cdot x + b_1, \ldots, a_m \cdot x + b_m) \mid a_i \in \{0,1\}^n, b_i \in \{0,1\}\}$$

is a family of pairwise independent hash functions.

Unique Solution from Hashing

Lemma 3

If $T \subseteq \{0,1\}^n$ such that $2^k \leq |T| < 2^{k+1}$ and $\mathcal{H}$ is a family of pairwise independent hash functions $h : \{0,1\}^n \to \{0,1\}^{k+2}$ then

$$\Pr_{h \in \mathcal{H}} [|\{x \in T \mid h(x) = 0\}| = 1] \geq 1/8$$

Unique Solution from Hashing

Lemma 3

If $T \subseteq \{0,1\}^n$ such that $2^k \leq |T| < 2^{k+1}$ and $\mathcal{H}$ is a family of pairwise independent hash functions $h : \{0,1\}^n \to \{0,1\}^{k+2}$ then

$$\Pr_{h \in \mathcal{H}} [|\{x \in T \ | \ h(x) = 0\}| = 1] \geq 1/8$$

- Fix $x \in T$. Want to compute

$$\Pr_h [h(x) \wedge \ \forall y \in T \setminus \{x\}, \ h(y) \neq 0]$$

Unique Solution from Hashing

Lemma 3

If $T \subseteq \{0,1\}^n$ such that $2^k \leq |T| < 2^{k+1}$ and $\mathcal{H}$ is a family of pairwise independent hash functions $h : \{0,1\}^n \rightarrow \{0,1\}^{k+2}$ then

$$\Pr_{h \in \mathcal{H}} [|\{x \in T \ | \ h(x) = 0\}| = 1] \geq 1/8$$

- Fix $x \in T$. Want to compute

$$\Pr_h[h(x) \wedge \ \forall y \in T \setminus \{x\}, \ h(y) \neq 0]$$

- Write

$$\Pr_h[\forall y \in T \setminus \{x\}, \ h(y) \neq 0 \ \mid h(x) = 0]$$
$$= 1 - \Pr[\exists y \in T \setminus \{x\} \text{ s.t. } h(y) = 0 \ \mid \ h(x) = 0]$$

Unique Solution from Hashing

- Union bound

$$
\begin{aligned}
&\Pr[\exists y \in T \setminus \{x\} \text{ s.t. } h(y) = 0 \mid h(x) = 0] \\
&= \sum_{y \in T \setminus \{x\}} \Pr[h(y) = 0 \mid h(x) = 0] \\
&= \sum_{y \in T \setminus \{x\}} \Pr[h(y) = 0]
\end{aligned}
$$

Construction of family of CNFs

Given a CNF ϕ, let S_ϕ be the set of satisfying assignments to ϕ.

Lemma 4

There is a (one-sided) poly-time PTM that on input a CNF formula ϕ and integer k outputs a formula ψ such that

1. *If ϕ is unsatisfiable then so is ψ* *(always)*
2. *If $2^k \leq |S_\phi| < 2^{k+1}$ then $|S_\psi| = 1$ with probability $\geq 1/8$*

Construction of family of CNFs

Lemma 4

There is a (one-sided) poly-time PTM that on input a CNF formula ϕ and integer k outputs a formula ψ such that

1. *If ϕ is unsatisfiable then so is ψ* *(always)*
2. *If $2^k \leq |S_\phi| < 2^{k+1}$ then $|S_\psi| = 1$ with probability $\geq 1/8$*

- Pick random $a_1, \ldots, a_{k+2} \in \{0,1\}^n$ and $b_1, \ldots, b_{k+2} \in \{0,1\}$

Construction of family of CNFs

Lemma 4

There is a (one-sided) poly-time PTM that on input a CNF formula ϕ and integer k outputs a formula ψ such that

1. *If ϕ is unsatisfiable then so is ψ* *(always)*
2. *If $2^k \leq |S_\phi| < 2^{k+1}$ then $|S_\psi| = 1$ with probability $\geq 1/8$*

- Pick random $a_1, \ldots, a_{k+2} \in \{0,1\}^n$ and $b_1, \ldots, b_{k+2} \in \{0,1\}$
- Will construct small CNF ψ which is equivalent to

$$\phi(x) \wedge (h_{a,b}(x) = 0) \Leftrightarrow \phi(x) \wedge \bigwedge_{i \in [n]} (a_i \cdot x + b_i = 0)$$

Challenge: $\bigoplus \notin AC^0_{/poly}$! How to write small CNF for $a_i \cdot x$?

Construction of family of CNFs

Lemma 4

There is a (one-sided) poly-time PTM that on input a CNF formula ϕ and integer k outputs a formula ψ such that

1. *If ϕ is unsatisfiable then so is ψ* *(always)*
2. *If $2^k \leq |S_\phi| < 2^{k+1}$ then $|S_\psi| = 1$ with probability $\geq 1/8$*

- Pick random $a_1, \ldots, a_{k+2} \in \{0,1\}^n$ and $b_1, \ldots, b_{k+2} \in \{0,1\}$
- Will construct small CNF ψ which is equivalent to

$$\phi(x) \wedge (h_{a,b}(x) = 0) \Leftrightarrow \phi(x) \wedge \bigwedge_{i \in [n]} (a_i \cdot x + b_i = 0)$$

- Auxiliary variables to the rescue! Let $y_1, \ldots, y_n$ new vars.

$$a \cdot x \oplus b \equiv \bigwedge_{i=1}^{n-1} (y_i = a_i \wedge x_i \oplus y_{i-1}) \wedge (y_n = a_n \wedge x_n \oplus y_{n-1} \oplus b)$$

Construction of family of CNFs

Lemma 4

There is a (one-sided) poly-time PTM that on input a CNF formula ϕ and integer k outputs a formula ψ such that

1. *If ϕ is unsatisfiable then so is ψ* *(always)*
2. *If $2^k \leq |S_\phi| < 2^{k+1}$ then $|S_\psi| = 1$ with probability $\geq 1/8$*

- Pick random $a_1, \ldots, a_{k+2} \in \{0,1\}^n$ and $b_1, \ldots, b_{k+2} \in \{0,1\}$
- Will construct small CNF ψ which is equivalent to

$$\phi(x) \wedge (h_{a,b}(x) = 0) \Leftrightarrow \phi(x) \wedge \bigwedge_{i \in [n]} (a_i \cdot x + b_i = 0)$$

- Auxiliary variables to the rescue! Let $y_1, \ldots, y_n$ new vars.

$$a \cdot x \oplus b \equiv \bigwedge_{i=1}^{n-1} (y_i = a_i \wedge x_i \oplus y_{i-1}) \wedge (y_n = a_n \wedge x_n \oplus y_{n-1} \oplus b)$$

- all expressions constantly many vars $\Rightarrow$ small CNF

Proof of Valiant-Vazirani

Theorem 5 (Valiant Vazirani 1986)

If there is poly-time algorithm which on input a CNF formula ϕ *with* $|S_\phi| = 1$ *finds the assignment, then RP* $=$ *NP.*

Proof of Valiant-Vazirani

Theorem 5 (Valiant Vazirani 1986)

If there is poly-time algorithm which on input a CNF formula ϕ *with* $|S_\phi| = 1$ *finds the assignment, then RP = NP.*

- Algorithm: run the below procedure 10 times
  1. On input ϕ use algorithm from Lemma 4 with parameters $0 \leq k \leq n$, constructing ψ_k for each k
  2. Accept if our poly-time algorithm finds a satisfying assignment to one of the formulas

Proof of Valiant-Vazirani

Theorem 5 (Valiant Vazirani 1986)

If there is poly-time algorithm which on input a CNF formula ϕ with $|S_\phi| = 1$ finds the assignment, then RP = NP.

- Algorithm: run the below procedure 10 times
  1. On input ϕ use algorithm from Lemma 4 with parameters $0 \leq k \leq n$, constructing ψ_k for each k
  2. Accept if our poly-time algorithm finds a satisfying assignment to one of the formulas
- If ϕ is unsatisfiable then procedure above will never accept

Proof of Valiant-Vazirani

Theorem 5 (Valiant Vazirani 1986)

If there is poly-time algorithm which on input a CNF formula ϕ *with* $|S_\phi| = 1$ *finds the assignment, then RP = NP.*

- Algorithm: run the below procedure 10 times
    1. On input ϕ use algorithm from Lemma 4 with parameters $0 \leq k \leq n$, constructing ψ_k for each k
    2. Accept if our poly-time algorithm finds a satisfying assignment to one of the formulas
- If ϕ is unsatisfiable then procedure above will never accept
- If ϕ is satisfiable, by lemma 4, each iteration of algorithm succeeds with probability $\geq 1/8$.
  Probability of success is $\geq 1 - (7/8)^{10} > 1/2$

References I

Arora, Sanjeev and Barak, Boaz (2009)
Computational Complexity, A Modern Approach — Chapter 17
Cambridge University Press

Trevisan, Luca (2002)
Lecture notes — Lecture 7
See webpage