Notes for Lecture 7

Today we see the Valiant-Vazirani result about the hardness of Unique-SAT.

1 The Valiant-Vazirani Reduction

In this section we show the following: suppose there is an algorithm for the satisfiability problem that always find a satisfying assignment for formulae that have exactly one satisfiable assignment (and behaves arbitrarily on other instances): then we can get an **RP** algorithm for the general satisfiability problem, and so **NP** = **RP**.

We prove the result by presenting a randomized reduction that given in input a CNF formula ϕ produces in output a polynomial number of formulae $\psi_0, \ldots, \psi_n$. If ϕ is satisfiable, then (with high probability) at least one of the ψ_i is satisfiable and has exactly one satisfying assignment; if ϕ is not satisfiable, then (with probability one) all ψ_i are unsatisfiable.

The idea for the reduction is the following. Suppose ϕ is a satisfiable formula with n variables that has about 2^k satisfying assignments, and let $h : \{0,1\}^n \to \{0,1\}^k$ be a hash function picked from a family of pairwise independent hash functions: then the average number of assignments x such that $\phi(x)$ is true and $h(x) = (0, \ldots, 0)$ is about one. Indeed, we can prove formally that with constant probability there is exactly one such assignment,[1] and that there is CNF formula ψ (easily constructed from ϕ and h) that is satisfied precisely by that assignment. By doing the above construction for values of k ranging from 0 to n, we obtain the desired reduction. Details follow.

Definition 1 *Let H be a family of functions of the form $h : \{0,1\}^n \to \{0,1\}^m$. We say that H is a family of pair-wise independent hash functions if for every two different inputs $x, y \in \{0,1\}^n$ and for every two possible outputs $a, b \in \{0,1\}^m$ we have*

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

Another way to look at the definition is that for every $x \neq y$, when we pick h at random then the random variables $h(x)$ and $h(y)$ are independent and uniformly distributed. In particular, for every $x \neq y$ and for every a, b we have $\mathbf{Pr}_h[h(x) = a | h(y) = b] = \mathbf{Pr}_h[h(x) = a]$.

For m vectors $a_1, \ldots, a_m \in \{0,1\}^m$ and m bits $b_1, \ldots, b_m$, define $h_{a_1,\ldots,a_m,b_1,\ldots,b_m} : \{0,1\}^n \to \{0,1\}^m$ as $h_{\mathbf{a},b}(x) = (a_1 \cdot x + b_1, \ldots, a_m \cdot x + b_m)$, and let H_{AFF} be the family of functions defined this way. Then it is not hard to see that H_{AFF} is a family of pairwise independent hash functions.

Lemma 1 *Let $T \subseteq \{0,1\}^n$ be a set such that $2^k \leq |T| < 2^{k+1}$ and let H be a family of pairwise independent hash functions of the form $h : \{0,1\}^n \to \{0,1\}^{k+2}$. Then if we pick*

[1] For technical reasons, it will be easier to prove that this is the case when picking a hash function $h : \{0,1\}^n \to \{0,1\}^{k+2}$.

h at random from H, there is a constant probability that there is a unique element $x \in T$ such that $h(x) = \mathbf{0}$. Precisely,

$$\mathbf{Pr}_{h \in H}[|\{x \in T : h(x) = \mathbf{0}\}| = 1] \geq \frac{1}{8}$$

PROOF: Let us fix an element $x \in T$. We want to compute the probability that x is the *unique* element of T mapped into $\mathbf{0}$ by h. Clearly,

$$\mathbf{Pr}_h[h(x) = \mathbf{0} \wedge \forall y \in T - \{x\}.h(y) \neq \mathbf{0}] = \mathbf{Pr}_h[h(x) = \mathbf{0}] \cdot \mathbf{Pr}_h[\forall y \in T - \{x\}.h(y) \neq \mathbf{0} | h(x) = \mathbf{0}]$$

and we know that

$$\mathbf{Pr}_h[h(x) = \mathbf{0}] = \frac{1}{2^{k+2}}$$

The difficult part is to estimate the other probability. First, we write

$$\mathbf{Pr}_h[\forall y \in T - \{x\}.h(y) \neq \mathbf{0} | h(x) = \mathbf{0}] = 1 - \mathbf{Pr}_h[\exists y \in T - \{x\}.h(y) = \mathbf{0} | h(x) = \mathbf{0}]$$

And then observe that

$$\begin{aligned}
& \mathbf{Pr}_h[\exists y \in T - \{x\}.h(y) = \mathbf{0} | h(x) = \mathbf{0}] \\
\leq & \sum_{y \in |T| - \{x\}} \mathbf{Pr}_h[h(y) = \mathbf{0} | h(x) = \mathbf{0}] \\
= & \sum_{y \in |T| - \{x\}} \mathbf{Pr}_h[h(y) = \mathbf{0}] \\
= & \frac{|T| - 1}{2^{k+2}} \\
\leq & \frac{1}{2}
\end{aligned}$$

Notice how we used the fact that the value of $h(y)$ is independent of the value of $h(x)$ when $x \neq y$.

Putting everything together, we have

$$\mathbf{Pr}_h[\forall y \in T - \{x\}.h(y) \neq \mathbf{0} | h(x) = \mathbf{0}] \geq \frac{1}{2}$$

and so

$$\mathbf{Pr}_h[h(x) = \mathbf{0} \wedge \forall y \in T - \{x\}.h(y) \neq \mathbf{0}] \geq \frac{1}{2^{k+3}}$$

To conclude the argument, we observe that the probability that there is a unique element of T mapped into $\mathbf{0}$ is given by the sum over $x \in T$ of the probability that x is the unique element mapped into $\mathbf{0}$ (all this events are disjoint, so the probability of their union is the sum of the probabilities). The probability of a unique element mapped into $\mathbf{0}$ is then at least $|T|/2^{k+3} > 1/8$. $\square$

Lemma 2 *There is a probabilistic polynomial time algorithm that on input a CNF formula ϕ and an integer k outputs a formula ψ such that*

- *If ϕ is unsatisfiable then ψ is unsatisfiable.*
- *If ϕ has at least 2^k and less than 2^{k+1} satifying assignments, then there is a probability at least $1/8$ then the formula ψ has exactly one satisfying assignment.*

PROOF: Say that ϕ is a formula over n variables. The algorithm picks at random vectors $a_1, \ldots, a_{k+2} \in \{0,1\}^n$ and bits $b_1, \ldots, b_{k+2}$ and produces a formula ψ that is equivalent to the expression $\phi(x) \wedge (a_1 \cdot x + b_1 = 0) \wedge \ldots \wedge (a_{k+2} \cdot x + b_{k+2} = 0)$. Indeed, there is no compact CNF expression to compute $a \cdot x$ if a has a lot of ones, but we can proceed as follows: for each i we add auxiliary variables $y_1^i, \ldots, y_n^i$ and then write a CNF condition equivalent to $(y_1^i = x_1 \wedge a_i[1]) \wedge \cdots \wedge (y_n^i = y_{n-1}^i \oplus (x_n \wedge a_i[n] \oplus b_i)))$. Then ψ is the AND of the clauses in ϕ plus all the above expressions for $i = 1, 2, \ldots, k+2$.

By construction, the number of satisfying assignments of ψ is equal to the number of satisfying assignments x of ϕ such that $h_{a_1,\ldots,a_{k+2},b_1,\ldots,b_{k+2}}(x) = \mathbf{0}$. If ϕ is unsatisfiable, then, for every possible choice of the a_i, ψ is also unsatisfiable.

If ϕ has between 2^k and 2^{k+1} assignments, then Lemma 1 implies that with probability at least $1/8$ there is exactly one satisfying assignment for ψ. $\Box$

Theorem 3 (Valiant-Vazirani) *Suppose there is a polynomial time algorithm that on input a CNF formula having exactly one satisfying assignment finds that assignment. (We make no assumption on the behaviour of the algorithm on other inputs.) Then* $\mathbf{NP} = \mathbf{RP}$.

PROOF: It is enough to show that, under the assumption of the Theorem, 3SAT has an **RP** algorithm.

On input a formula ϕ, we construct formulae $\psi_0, \ldots, \psi_n$ by using the algorithm of Lemma 2 with parameters $k = 0, \ldots, n$. We submit all formulae $\psi_0, \ldots, \psi_n$ to the algorithm in the assumption of the Theorem, and accept if the algorithm can find a satisfying assignment for at least one of the formulae. If ϕ is unsatisfiable, then all the formulae are always unsatisfiable, and so the algorithm has a probability zero of accepting. If ϕ is satisfiable, then for some k it has between 2^k and 2^{k+1} satisfying assignments, and there is a probability at least $1/8$ that ψ_k has exactly one satisfying assignment and that the algorithm accepts. If we repeat the above procedure t times, and accept if at least one iteration accepts, then if ϕ is unsatisfiable we still have probability zero of accepting, otherwise we have probability at least $1 - (7/8)^t$ of accepting, which is more than $1/2$ already for $t = 6$. $\Box$

Exercises

1. Prove that H_{AFF} is indeed a family of a pair-wise independent hash functions.
2. Change the assumption of Theorem 3 to having a *probabilistic* polynomial time algorithm that on input a formula with exactly one satisfying assignment finds that assignment with probability at least $1/2$. Prove that it still follows that $\mathbf{NP} = \mathbf{RP}$.