6.841 Advanced Complexity Theory May 7, 2007

Lecture 23

Lecturer: Madhu Sudan *Scribe: Alex Andoni, Anastasios Sidiropoulos*

1 Overview

Topics for this lecture are:

- Continue the discussion on Average-Case analysis (as opposed to Worst Case);
- Present Impagliazzo's five possible worldsl

Administrativia:

- Project presentations are on Wed and Thu in 32-G531 and 32-G631 respectively;
- Email comments on PCP and Average-Case lectures by Tuesday.
- Fill out HKN Survey online.

2 Literature on Average-Case Complexity

Some of important surveys on average case complexity (very far from being all of them) are, in approximate chronological order:

- Levin [1] formalized the idea that average case complexity is about problems *plus a distribution* over inputs (i.e., hardness depends on the the distribution).
- Impagliazzo [2] wrote a survey giving his "Personal View of Average-Case Complexity" describing 5 possible worlds (we live in exactly one of them; we just don't know which one, yet).
- Goldreich wrote a survey, that made its way into his book [3].
- Ajtai [4] gave a talk at ICM'02 on connections between Worst-Case complexity and Average-Case Complexity, specifically, in the context on lattice problems.
- Bogdanov and Trevisan [5] recently wrote a survey on "Average-Case Complexity".

In this lecture, we discuss Impagliazzo's five possible worlds, as well as Ajtai's lattice problems.

3 Impagliazzo's five possible worlds

Russell Impagliazzo wrote a survey on Average-Case Complexity [2] describing 5 possible worlds: we live in one of them, but do not yet know which one.

The motivation for the classification is to relate cryptography to worst-case/average-case complexity. A question raised by Diffie-Hellmann was whether we can base cryptography on strong assumptions such as P $\neq$ NP. Today we can't, and there are roughly 3 questions that, at the moment, seem relatively independent:

- P $\neq$ NP;
- Existence of one-way functions (defined below). This implies some cryptography (Diffie-Hellmann's protocol);

- Existence of Public Key CryptoSystems (PKCS) (best example of it is, of course, RSA).

The only implications we know are that PKCS implies existence of one-way functions, which, in turn, imply P $\neq$ NP. Where does the truth lie?

Definition 1 *A* one-way function *is a function $f : \{0,1\}^* \rightarrow \{0,1\}^*$ such that it is easy to compute but hard to invert on average, i.e.:*

Easy: Computing $f(x)$ takes poly($|x|$) time;

Hard: Given a random $x \leftarrow U_n$ (uniform over $\{0,1\}^n$), it is hard to invert $f(x)$, that is for any poly-time probabilistic-time algorithm A, we have that

$$\Pr_{x \leftarrow U_n}[A(f(x)) \in f^{-1}(f(x))] = \text{neglijible}.$$

(Note that it would be too harsh to require that A returns x precisely since $f(x)$ migth have simply lost this information.)

These considerations led to the definitions of DNP and Avg − BPP.

Impagliazzo's five worlds roughly assume different scenarios of validity of the above questions.

To put worlds in a perspective, he uses the story of Gauss's class in school, and his Professor Grouse. The idea is that Professor Grouse wants to humiliate Gauss in front of the class by giving him (inventing) a problem that Gauss cannot solve. In each of the 5 worlds, we will see whether Professor manages to humiliate Gauss or not. In these considerations, we assume that both Professor Grouse and Gauss are poly-time algorithms[1].

1. [**Algorithmica**] The world where P = NP. Then we have efficient algorithms for all NP-complete problems. Professor Grouse cannot embarass Gauss since Gauss can efficiently solve all problems that Professor Grouse can give him (well, at least from the class of problem that do have an efficiently describable solution/proof).

2. [**Heuristica**] P $\neq$ NP but DNP $\subseteq$ Avg − BPP. In some sense there are hard instances to NP-complete problems, but it's hard to find those instances (they are not poly-time sample-able).

   Professor Grouse cannot humiliate Gauss by giving a hard problem (unless he spends a lot of time looking for such problem, say, polynomially more time than Gauss uses to solve it).

   Note that this world is still different from Algorithmica, even "in practice" (see Impagliazzo paper for details).

3. [**Pessiland**] In this world, the world of a pessimist, DNP $\not\subseteq$ Avg − BPP but one-way functions don't exist. The first condition says it is easy to come up with hard instances (there are poly-time sampleable distributions that are hard). However, the second condition says that it is hard to generate hard instances together with the answers (since we can invert the generation process and find the random bits used to generate problem+solution).

   The implication for Professor Grouse is that he can humiliate Gauss but in a very restricted sense. Professor Grouse can give Gauss a hard problem, but he will not know the answer to his own question!

   Although in this "grey world", NP-complete problems are hard, and crypto does not exist, there are some positive implications.

4. [**Minicrypt**] In this world, one-way functions exist but not PKCS. Some limited form of cryptography is possible.

   Professor Grouse can finally humiliate Gauss: Professor Grouse can generate problems, together with solutions, such that Gauss cannot solve these problems.

[1] Impagliazzo leaves it unspecified whether these are deterministic or randomized or even quantum algorithms.

5. [**Cryptomania**] PKCS exist, and we can regain some confidence in e-Commerce.

   Professor Grouse can stage a superior form of humiliation, in addition to the above form of humiliation. Professor Grouse can choose the "dumbest" student in the class (on the fly), and then, generate a problem that Gauss cannot solve but this dumbest student *can* solve. All this happens in a public setting (no private channel between the professor and the dumbest student).

The last world seems most believable.

4 Random 3SAT

There are some "empirical" approaches to solving hard problems (NP) in average-case (i.e., take a problem and solve on "random" instances). However, so far, people have tried and failed. One of the illustrative directions is Random 3SAT problem.

For a 3CNF formula ϕ with m clauses on n variables, we define the *density* of ϕ to be the ratio $\Delta(\phi) = m/n$. For each $\delta > 0$, $n > 0$, we define a distribution $\mathcal{D}_{\delta,n}$ on 3CNF formulae on n variables with density δ as follows. For a given number of variables n, we pick $m = \delta \cdot n$ clauses, where for each clause we pick 3 literals uniformly at random. The resulting average-case computational problem is referred to as *Random 3SAT*.

The study of the behavior of Random 3SAT has attracted a lot of attention in Statistical Physics, and Probability Theory in general.

It is easy to see that for density δ close to 0, the probability that a clause in $\mathcal{D}_{\delta,n}$ tends to 1. Moreover, for density δ close to ∞, this probability tends to 0. In fact, it has been conjectured that there exists a phase transition on the fraction of satisfiable formulae of density δ, around some density $\Delta_0 \approx 4.2$. More precisely,

Conjecture 1 *There exists* $\Delta_0 \approx 4.2$*, such that*

- *For each* $\Delta < \Delta_0$, $\lim_{n\to\infty} \mathbf{Pr}_{\phi\in\mathcal{D}_{\Delta,n}}[\phi \textit{ is satisfiable}] = 1$.
- *For each* $\Delta > \Delta_0$, $\lim_{n\to\infty} \mathbf{Pr}_{\phi\in\mathcal{D}_{\Delta,n}}[\phi \textit{ is satisfiable}] = 0$.

Note that if the above conjecture is true, then Random 3SAT is easy for all densities $\Delta \neq \Delta_0$: one can simply output YES if the density of the given formula is greater than Δ_0, and NO otherwise. In fact, all (heuristic) algorithms that are currently known for Random 3SAT fail for densities close to Δ_0. This fact lead to the formulation of the following computational conjecture.

Conjecture 2 *If we pick a random 3CNF formula ϕ of density Δ_0, then it is hard to tell if ϕ is satisfiable.*

The above conjecture implies that DNP $\subsetneq$ Avg-BPP. On the other hand, one could also formulate an opposite conjecture:

Conjecture 3 *There exists a polynomial-time algorithm A, and a family of formulae $\{S_n\}_{n\in\mathbb{N}}$, such that*

- $\mathbf{Pr}_{\phi\in\mathcal{D}_{\Delta_0,n}}[\phi \in S_n]$ *is negligible.*
- *For each* $\phi \notin S_n$, $A(\phi) = SAT(\phi)$.

Note that conjectures 1 and 3 can be simultaneously true. That is, a phase transition in Random 3SAT might not necessarily imply a computational hardness on average for density Δ_0. In fact, Ben-Sasson has shown that there exists an NP-hard problem that gives rise to a phase transition with respect to some appropriate parameter, yet it is easy on average for every possible value of this parameter.

5 Shortest Vector Problem

Ajtai used problems based on lattice to introduce a worst-case problem Π_1 which is not known to be in P, that can be reduced to a distributional problem (Π_2, D).

A lattice L in $\mathbb{Q}^n$ is defined as a discrete additive subset of $\mathbb{Q}^n$. That is, for each $x, y \in L$, $x + y \in L$, and $x - y \in L$, and also there exists $\epsilon > 0$, s.t. for each $x \in L$, $\mathrm{Ball}(x, \epsilon) \cap L = \{x\}$.

The length of the shortest vector in L, denoted by $\lambda_1(L)$ is defined as the minimum distance between two elements in L.

A lattice in $\mathbb{Q}^n$ can be given represented by a set of vectors $b_1, \ldots, b_n \in \mathbb{Q}^n$, such that

- $b_1, \ldots, b_n$ are linearly independent.
- $L = \{\sum_{i \in [n]} r_i \cdot b_i | r \in \mathbb{Z}^n\}$.

Therefore, with the above representation a lattice can be given as input to an algorithm. The computational problem SVP of finding $\lambda_1(L)$ is notoriously hard.

One can define an approximate version of SVP, called Approx-SVP$_g$ as follows. Given a lattice L by its basis $(b_1, \ldots, b_n)$, output a non-zero vector $v \in L$, s.t. $\|v\|_2 \leq g(n) \cdot \lambda_1(L)$.

It has been shown that for each polynomial g_2, there exists a polynomial g_1, such that the worst-case problem Approx-SVP$_{g_1}$ can be reduced to the distributional problem (Approx-SVP$_{g_2}$, D_{g_2}).

Moreover, Coldreich and Coldwasser have shown that the problem Approx-SVP cannot be NP-hard.

Feigenbaum and Fortnow, and Bogdanov and Trevisan have shown that if there exists a many-one reduction from SAT to the same DNP problem (Π, D), then PH collapses.

References

[1] Leonid Levin. Average Case Complete Problems. SIAM J. Comp. 15 (1986), pp. 285-286.

[2] Russell Impagliazzo. A Personal View of Average-Case Complexity. 1995.

[3] Oded Goldreich. *Computational Complexity: A Conceptual Perspective.* In preparation. Available at `http://www.wisdom.weizmann.ac.il/ oded/cc-book.html`.

[4] Miklós Ajtai. Worst-Case Complexity, Worst-Case Complexity and Lattice Problems. *Doc. Math. J. DMV* III, pp. 421-428. Extra Volume ICM 1998.

[5] Andrej Bogdanov and Luca Trevisan: Average-case complexity. In *Foundations and Trends in Theoretical Computer Science*, volume 2, issue 1, Now Publishers, 2006.