Differential Privacy Study Group **May 6, 2016**

PCP Theorem

Leader: KM Chung **Notes: Chi-Ning Chou**

This week we are going to use PCP theorem to show the nonexistence of efficient DP mechanism outputing synthesized dataset for all 2-way marginals.

In this note, we will focus on the PCP theorem. First, we will quickly have an overview on the historical development of PCP theorem and its relation with interactive proof in Section 1. Next, we will formally state the PCP theorem in Section 2 and discuss the underlying intuition. Finally, in Section 3, one of the most important application of PCP theorem will be introduced: *inapproximability*.

1 Historical development

> *"What is intuitively required from a theorem-proving procedure? First, that it is possible to prove a true theorem. Second, that it is impossible to prove a false theorem. Third, that commu- nicating the proof should be efficient, in the following sense. It does not matter how long must the prover compute during the proving process, but it is essential that the computation required from the verifier is easy."*
>
> Goldwasser, Micali, Rackoff 1985

1.1 Interactive proof system

Interactive proof system was proposed by Goldwasser, Micali, Rackoff in 1980s[GMR89]. In the system, there are an unbounded prover $\mathbf{P}$ and a computationally bounded verifier $\mathbf{V}$. During the computation, the verifier receives the problem and can ask some questions to the prover. The prover then do some hidden computation and return the answer back to the verifier. In this process, we require the time running by the verifier to be computationally efficient. Moreover, since sometimes we allow randomness, we also need to put some constraint on the completeness and soundness. We call a certain pair of verifier and prover with a communication protocol $\langle \mathbf{P}, \mathbf{V} \rangle$ an interactive proof system.

Intuitively, interactive proof system proposed a new concept of **proving** a problem. Instead of directly compute the answer on one's own, with the setting of interactive proof, the verifier are **convinced** by the prover with the communication protocol set in advance.

Formally, we can define a natural complexity class for interactive proof system where the verifier runs in polynomial time as follow.

Definition 1 (IP) *If problem $L \in \mathbf{IP}$, then there exists $\langle \mathbf{P}, \mathbf{V} \rangle$ such that for any input $x \in \{0,1\}^*$, $\mathbf{V}$ can run in poly$(|x|)$ time and*

- *(Completeness): If $x \in L$, then $\mathbb{P}[\langle \mathbf{P}, \mathbf{V} \rangle(x) = 1] = 1$*
- *(Soundness) If $x \notin L$, then $\mathbb{P}[\langle \mathbf{P}, \mathbf{V} \rangle(x) = 0] \geq \frac{1}{2}$. Moreover, for any malicious prover $\mathbf{P}'$, we still have $\mathbb{P}[\langle \mathbf{P}', \mathbf{V} \rangle(x) = 0] \geq \frac{1}{2}$.*

There are a bunch of wonderful results about interactive proof system, however, they are not the main focus of today, so I will only list several benchmark results. First, Graph-Nonisomorphism, which is unknown to be in **NP**, is in **IP**. Furthermore, **IP** = **PSPACE**, which is an important nonrelativized result.

1.2 Probabilistic checkable proof

Now, let's turn to today's main topic: *probabilistic checkable proof (PCP)*. Similar to interactive proof system, in PCP, we also have a computationally bounded verifier $\mathbf{V}$ while on the other hand, there's no prover. Instead, there's a proof π which can be accessed by the verifier. For problem instance x, the verifier can use $r(|x|)$ randomness to access $q(|x|)$ bits in the proof π. Then, after running its computation in polynomial time, the verifier will decide to accept or reject x. Formally, we define the following family of complexity class.

Definition 2 ($\mathbf{PCP}(r(n), q(n))$) *We say $L \in \mathbf{PCP}(r(n), q(n))$ if there exists a probabilistic polynomial time (PPT) verifier $\mathbf{V}$ such that for any input x, $\mathbf{V}$ can use $r(|x|)$ randomness and have oracle access to $q(|x|)$ bits of a proof π and*

- *(Completeness): If $x \in L$, then $\mathbb{P}[\mathbf{V}^{\pi}(x) = 1] = 1$.*
- *(Soundness): If $x \notin L$, then $\mathbb{P}[\mathbf{V}^{\pi}(x) = 0] \geq \frac{1}{2}$. Moreover, for a problematic proof π', we still have $\mathbb{P}[\mathbf{V}^{\pi'}(x) = 0] \geq \frac{1}{2}$.*

Intuitively, the definition of **PCP** is highly connected to **NP**, which is a complexity class containing problems that can be certified in polynomial time. However, there are two main differences here.

1. Randomness. In **NP**, all the setting include the verification does not involve any randomness while **PCP** allows certain amount of randomness used in the verification process.
2. Access of proof π. In **NP**, we do not regulate the use of π while in **PCP**, the verifier might only access a small portion of proof.

At first glance, it seems that the above two differences of **NP** and **PCP** might lead to totally different behaviors. Surprisingly, the PCP theorem gave a direct connection between these two complexity classes: $\mathbf{NP} = \mathbf{PCP}(O(\log(n)), O(1))$.

2 PCP theorem

Theorem 3 (The PCP theorem, [AS98][ALM+98])
$\mathbf{NP} = \mathbf{PCP}(O(\log(n)), O(1))$

Basically, the PCP theorem told us one thing:

Deterministically access a poly-length proof

$\equiv$

Using logarithmic randomness to access constant bits in the proof

PCP theorem provided a brand new aspect of viewing **NP**. With the probabilistic property of **PCP**, when the input x is not in the problem L, there's nonzero probability for the verifier to accept x. On the other hand, take a look at the soundness of **NP**, the verification process must reject x as long as it does not in L. How can these two seemingly distinct correctness notion shown to be equivalent? Moreover, PCP theorem also told us that only constant number of bits in π are required!

Here, we won't discuss the proof of PCP theorem today since it takes weeks to go through all the details. Nevertheless, to convince yourself the correctness of PCP theorem, one can imagine that the prover in **PCP** in some sense utilize the randomness to traverse every bit in the proof π. Although he cannot access every bit in a single realization, he can somehow guarantee that if $x \in L$, in every possible combination of proof bits, he can certify $x \in L$. On the other hand, if $x \notin L$, there's at least half of the possible combination of proof bits can reveal the fact that $x \notin L$.

2.1 Resource

For more information and proof about PCP theorem, there are nice lecture by Venkatesan Guruswami and Ryan O'Donnell: *CSE 533: The PCP Theorem and Hardness of Approximation, Autumn 2005*. In addition, chapter 11 in [AB09] is also a nice material.

3 Inapproximability

Finally, it's time for us to see the main application of PCP theorem: inapproximability. To learn the inapproximability of problems, we need to first understand the subject we want to approximate: *NP optimization problem*. After that, we can interpret the PCP theorem in a totally different aspects and see its interesting application.

3.1 NP optimization problem

There are lots of common **NP**-complete problem having analogous optimization problem such as Clique, 3Sat, Max-Cut, etc. As long as we don't know how to directly solve the decision version, we cannot easily find the optimum either. As a result, we might want to instead just **approximate** the answer. Namely, for Clique we might want to approximate the maximum size of the clique in the given graph, for 3Sat we might want to approximate the maximum number of satisfying clauses, etc. Formally, we define the approximation notion as follow.

Definition 4 ($(1 \pm \epsilon$-approximation)) *For a maximization(minimization) problem L. We say $A(\cdot)$ is an $(1-\epsilon)$-approximation ($(1+\epsilon)$-approximation) for L if for any input x with optimum value $OPT(x)$, $A(x) \geq (1-\epsilon)OPT(x)$ ($A(x) \leq (1+\epsilon)OPT(x)$).*

For an **NP**-complete optimization problem, we would like to have a polynomial time $(1\pm)$-approximation for ϵ as small as possible. If a optimization problem have polynomial time ϵ-approximation for any $\epsilon > 0$, then we say it has a *polynomial time approximation scheme (PTAS)*.

Surprisingly, it turns out that some **NP**-complete problem has PTAS while most of them don't have. For example, KNAPSACK has PTAS while there exists $\epsilon > 0$ such that it is **NP**-hard to have an $(1-\epsilon)$-approximation for MAX-3SAT.

3.2 Inapproximability interpretation of PCP theorem

From the original definition of PCP, we view it as a proof system. Namely, the PCP theorem provided another characterization of **NP**. Here, we are going to give a different interpretation of PCP theorem in the context of inapproximability.

Before we formally state and prove the new interpretation of PCP theorem, let's recall the gap version problem.

For an **NP** optimization problem L, we can define its gap version as follow.

Definition 5 (ρ-GAP-L) *For any input x of problem ρ-GAP-L, x. Let val be the valuation function of ρ-GAP-L, we have*

- *If $x \in \rho$-GAP-L, then $val(x) = 1$.*
- *If $x \notin \rho$-GAP-L, then $val(x) < \rho$.*

That is, a yes instance will always maximize the valuation function to 1 (e.g. satisfy all clauses in MAX-3SAT) while a no instance must have value less than ρ. Note that this is a promise decision problem for **NP** optimization problem. Intuitively, if it is hard to solve a ρ-GAP-L problem, then it will be hard to approximate problem L with rate ρ.

It turns out that PCP theorem is actually equivalent to saying that there exists a constant ρ for the gap version of some **NP** optimization such that it is **NP**-hard to solve the gap problem. As a result, it is **NP**-hard to approximate the optimization problem within ρ factor!

3.3 MAX-3SAT has no PTAS

Now, we are going to see how to derive inapproximability result from the proof version of PCP theorem stated in Theorem 3. First, let's see the formal statement of the inapproximability of MAX-3SAT.

Theorem 6 *$\exists \epsilon > 0$ such that $(1-\epsilon)$-approximation for* MAX-3SAT *is* **NP***-hard.*

Intuitively, the idea is to reduce a PCP process of an **NP**-complete problem to the gap version of MAX-3SAT. For example, if one can show that deciding $(1-\epsilon)$-GAP-MAX-3SAT is equivalent to having a PCP for some **NP**-complete problem with logarithmic randomness and constant access to a proof π with desired correctness, then from Theorem 3, we know that it is **NP**-hard to decide $(1-\epsilon)$-GAP-MAX-3SAT. That is, $(1-\epsilon)$-approximation for MAX-3SAT is **NP**-hard.

PROOF: Given a **NP**-complete problem L, from Theorem 3, we know that there exists a probabilistic checkable proof with $r(n)$ randomness and q bits access to a length $l(n)$ proof, where $r(n) = O(\log n)$ and $l(n) = poly(n)$. Given an input x of length n and a proof π of length $l(n)$, from the correctness criteria of PCP theorem, we have

- If $x \in L$, $\mathbb{P}_{r \xleftarrow{\$} \{0,1\}^{r(n)}}[\mathbf{V}^{\pi}(x) = 1] = 1$.
- If $x \notin L$, $\mathbb{P}_{r \xleftarrow{\$} \{0,1\}^{r(n)}}[\mathbf{V}^{\pi}(x) = 1] < \frac{1}{2}$.

Let $val(\phi)$ be the maximum portion of clauses being satisfied in ϕ, our goal is to reduce the above proof process to a gap instance ϕ_x of $(1-\epsilon)$-Gap-Max-3Sat where ϵ is a constant decided later. That is, ϕ_x should satisfy the gap property.

- If $\phi_x \in (1-\epsilon)$-Gap-Max-3Sat, then $val(x) = 1$.
- If $\phi_x \notin (1-\epsilon)$-Gap-Max-3Sat, then $val(x) \leq 1 - \epsilon$.

To construct ϕ_x, first observe that in the PCP process above, if $x \in L$, then for all the possible randomness $r \in \{0,1\}^{r(n)}$, after accessing q corresponding bits $\pi(i_1(r)), \dots \pi(i_q(r))$, $\mathbf{V}$ must accept it. Intuitively, it's like satisfying all the possible constant-size realization of verification process. On the other hand, if $x \notin L$, then there are at least half of the randomness $r \in \{0,1\}^{r(n)}$ failing to convince the verifier and thus end up with a rejection. Similarly, it's like a constant portion of realization in the verification process is rejected.

Concretely, in the PCP process above, for any possible randomness realization $r \in \{0,1\}^{r(n)}$, we can view the verification process as a function $V_{r,x} : \{0,1\}^q \to \{0,1\}$. By Cook-Levin theorem, we can simply have a 3SAT instance $\phi_{r,x}(y_{i_1}, \dots, y_{i_q}, z_1, \dots, z_q)$ where $y_{i_1}, \dots, y_{i_q}$ are corresponded to the proof bits we used and $z_1, \dots, z_q$ are just dummy variables. Note that the number of clauses in $\phi_{r,x}$ is at most q^{2^q}. By the correctness of PCP theorem, we have

- If $x \in L$, then $\exists \pi$ such that $\forall r \in \{0,1\}^{r(n)}$, $\phi_{r,x}(\pi(i_1(r)), \dots \pi(i_q(r)), z_{r,1}, \dots, z_{r,q}) = 1$ with some assignment on the dummy variables.
- If $x \notin L$, then $\forall \pi \in \{0,1\}^{l(n)}$, there are half of the randomness r in $\{0,1\}^{r(n)}$ such that $\phi_{r,x}(\pi(i_1(r)), \dots, \pi(i_q(r)), z_{r,1}, \dots, z_{r,q}) = 0$ for all possible dummy assignment.

As we can view the whole verification process as a large function $V : \{0,1\}^{l(n)} \to \{0,1\}$ where $V(\pi(1), \dots, \pi(l)) = \mathbf{1}_{\{\forall r \in \{0,1\}^{r(n)}, V_{r,x}(\pi(i_1(r)), \dots, \pi(i_q(r))) = 1\}}$, it is actually equivalent to consider the 3SAT $\phi_x(y_1, \dots, y_{l(n)}, \bar{z}) = \wedge_{r \in \{0,1\}^{r(n)}} \phi_{r,x}(y_{i_1(r)}, \dots, y_{i_q(r)}, z_{r,1}, \dots, z_{r,q})$ which has at most $2^{r(n)} q^{2^q}$ clauses. Let $\epsilon = \frac{1}{2q^{2^q}}$. Finally, from the above correctness observation, we have

- If $x \in L$, then ϕ_x is satisfiable. Namely, $\phi_x \in (1-\epsilon)$-Gap-Max-3Sat.
- If $x \notin L$, then for any proof π, there are at least half of the randomness $r \in \{0,1\}^{r(n)}$ such that $V_{r,x}$ rejects. That is, for any assignment $y_1, \dots, y_{l(n)}, \bar{z}$, at least one clause in $\phi_{r,x}$ is not satisfiable. Namely, there are at least $2^{r(n)}/2$ clauses in ϕ_x is not satisfiable. As ϕ_x has at most $2^{r(n)} q^{2^q}$ clauses, we know that $\phi_x \notin (1-\epsilon)$-Gap-Max-3Sat.

To sum up, we have shown that $x \in L \Leftrightarrow \phi_x \in (1-\epsilon)$-Gap-Max-3Sat. By Theorem 3, we conclude that $(1-\epsilon)$-Gap-Max-3Sat is **NP**-hard. □

In Table 1, we summarize the correspondence between the proof perspective and the inapproximability perspective.

Proof system		Inapproximability
Verifier $\mathbf{V}$	$\Leftrightarrow$	Problem instance
Proof π	$\Leftrightarrow$	Assignment
Length of the proof	$\Leftrightarrow$	Number of variables
Number of queries q	$\Leftrightarrow$	Arty of constraints
Number of randomness	$\Leftrightarrow$	Logarithm of number of constraints
Soundness parameter	$\Leftrightarrow$	Maximum of $val(x)$ for a no instance

Table 1: Two interpretations of PCP theorem.

Basically, the whole reduction only contains the following steps:

$$
\begin{gathered}
x \in L \\
(x \notin L) \\
\Updownarrow \\
\mathbb{P}_{r \xleftarrow{\$} \{0,1\}^{r(n)}}[\mathbf{V}^{\pi}(x) = 1] = 1 \\
(\mathbb{P}_{r \xleftarrow{\$} \{0,1\}^{r(n)}}[\mathbf{V}^{\pi}(x) = 1] < \tfrac{1}{2}) \\
\Updownarrow \\
\exists \pi \text{ such that } \forall r \in \{0,1\}^{r(n)}, \\
(\forall \pi \in \{0,1\}^{l(n)}, \text{ half of } r \text{ in } \{0,1\}^{r(n)} \text{ such that} \\
\phi_{r,x}(\pi(i_1(r)), \ldots, \pi(i_q(r)), z_{r,1}, \ldots, z_{r,q}) = 0) \\
\Updownarrow \\
\phi_x \in (1-\epsilon)\text{-Gap-Max-3Sat} \\
(\phi_x \notin (1-\epsilon)\text{-Gap-Max-3Sat})
\end{gathered}
$$

Remark: There's an improvement from Håstad that approximate $(\frac{7}{8}+\epsilon)$-Max-3Sat is **NP**-hard, which is optimal since we have a efficient approximation for $\frac{7}{8}$-Max-3Sat.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of np. *Journal of the ACM (JACM)*, 45(1):70–122, 1998.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989.