Chapter 5
PCP Theorem and the Hardness of Approximation

Angsheng Li

Institute of Software
Chinese Academy of Sciences

Advanced Algorithms, U CAS
28th, March, 2016

Outline

1. Backgrounds
2. Approximation
3. PCP
4. Equivalence of two views
5. Linearity test
6. Fourier analysis
7. Exercises

Probabilistically checkable proofs (PCP)

The PCP theorem:

1. A gap reduction
2. hardness of approximation
3. new definition of NP
4. new methods and ideas for algorithms

Locally testable proof systems

Approach I: Locally testable proof systems
For a language L in NP, **Verification is easy**

PCP definition of NP: There is a verifier V,
Completeness: if $x \in L$, for the certificate σ of $x \in L$, encode σ to a codeword π, V randomly checks 3 bits of π, with prob ≈ 1, accepts.
Soundness If $x \notin L$, for any claimed certificate σ, encode it to π, V randomly accesses 3 bits of π, with prob $< \frac{1}{2}$, accepts.
Verification is even much easier

Hardness of approximation

- **Approach II** Hardness of approximation
  For many NP-hard optimization problem, finding a good approximate solution is as hard as finding the optimum solution
- **Conclusion**: The two approaches are equivalent.

Approximation view

Definition

(Approximation of MAX 3SAT) Given a 3CNF formula ϕ, the *value* of ϕ, denoted by $\mathrm{val}(\phi)$, is the maximum fraction of clauses that are satisfied by an assignment to ϕ's variables. For $\rho \leq 1$, we say that an algorithm *A* is ρ-approximation algorithm for MAX-3SAT, if for every 3CNF formula ϕ with *m* clauses, *A* on input ϕ runs in poly time, outputs and assignment that satisfies at least $\rho\mathrm{val}(\phi)m$ clauses of ϕ.

Recall:
1) Semidefinite programming leads to $\frac{7}{8} - \epsilon$ approximation algorithm for MAX-3SAT.
2) Min-Vertex cover VC has $\frac{1}{2}$ approximation algorithm.

Locally testable proofs- I

Remark: Not IP
Take SAT for example

1. Alice wants to convince Bob
   1.1 a CNF ϕ is satisfied
   1.2 Alice presents an assignment σ of ϕ to Bob
2. Bob checks if σ satisfies ϕ
   However, Bob has to check the whole σ in time polynomial of n.

Locally testable proofs- II

The new view is:

- Using an ECC, let $\pi = E(\sigma)$
- Bob checks only a constant bits of π
- With high probability, Bob is correct

Why?

i) the proof π is robust

ii) a few bits of π determines the property of the assignment σ – again, why this is possible

iii) Food test?

Verifier for PCP - informal

In a PCP, the verifier V satisfies:

(1) V is probabilistic

(2) V has random access to a proof string π, i.e., V can get each bit b_i of π by query, for V only needs to know the locations i for π_i
- Number of queries
- answer size, the length of a symbol π_i, if π is not $0, 1$ string

(3) V runs in poly time

(4) adaptive or nonadaptive

PCP verifier - formal definition

Let L be a language, $q, r : \mathbb{N} \to \mathbb{N}$. We say that L has an $(r(n), q(n))$-PCP verifier, if there is a polynomial time probabilistic algorithm V satisfying:

Efficiency: On an input string $x \in \{0,1\}^n$, and given random access to a proof string $\pi \in \{0,1\}^*$ of length at most $q(n)2^{r(n)}$, V uses at most $r(n)$ random coins, and makes at most $q(n)$ nonadaptive queries to locations of π. Output $V^\pi(x)$.

Completeness If $x \in L$, then $\exists \pi$,

$$\Pr[V^\pi(x) = 1] = 1$$

Soundness If $x \notin L$, then $\forall \pi$

$$\Pr[V^\pi(x) = 1] \leq \frac{1}{2}.$$

The PCP theorem: Locally testable proofs

PCP (r(n),q(n)): The languages having $(O(r(n)), O(q(n))$-PCP verifiers.

Theorem

$$\mathrm{NP} = \mathrm{PCP}(\log n, 1).$$

Remarks:

(1)
$$\mathrm{PCP}(r(n), q(n)) \subseteq \mathrm{NTIME}(2^{O(r(n))} \cdot q(n))$$

$$\mathrm{PCP}(\log n, 1) \subseteq \mathrm{NP}$$

(2) The number of queries is a universal constant.

(3) The soundness $\frac{1}{2}$ can be arbitrarily small such as $\frac{1}{2^c}$ for some c.

The PCP theorem: Hardness of approximation

Theorem

There exists a $\rho < 1$ such that for every $L \in \mathrm{NP}$, there is a polynomial time reduction R mapping strings to 3CNF formulas such that

$$x \in L \Rightarrow \mathrm{val}(R(x)) = 1$$

$$x \notin L \Rightarrow \mathrm{val}(R(x)) < \rho.$$

Remarks

1.
$$\exists \rho < 1 \forall L \exists R_L$$
– ρ is a universal constant for all L's, while each L has its own reduction R_L
2. Cook reduction
$$x \notin L \Rightarrow \mathrm{val}(R(x)) < 1.$$
– a surprising extension to Cook's theorem
– crucial idea is the gap given by $\rho < 1$, which is hence called a *gap reduction*

Hardness of approximation

Theorem

There exists a constant $\rho < 1$ such that if there is a polynomial time ρ-approximation algorithm for MAX 3SAT, then $\mathrm{P} = \mathrm{NP}$.

Proof.

Let A be an algorithm such that for any ϕ, A on input ϕ outputs an assignment that satisfies $\rho \cdot \mathrm{OPT}(\phi)$ clauses.
For an NP language L, and instance x,
If $x \in L$, $R(x)$ is satisfied, A on input $R(x)$ is an assignment satisfies ρm clauses.
If $x \notin L$, $\mathrm{val}(R(x)) < \rho$, so $\mathrm{OPT} < \rho m$. $A(R(x))$ satisfies at most $\mathrm{OPT} < \rho m$ clauses.
$x \in L$ if and only if $A(R(x))$ satisfies at least ρm clauses.
$\mathrm{NP} \subseteq \mathrm{P}$.

$\square$

Does Cook reduction help?

Cook reduction R.

$$L \in \mathrm{NP} \Rightarrow \mathrm{3CNF}$$

$$x \in L \Rightarrow \mathrm{val}(R(x)) = 1$$

$$x \notin L \Rightarrow \mathrm{val}(R(x)) < 1$$

Gap reduction requires:

$$x \notin L \Rightarrow \mathrm{val}(R(x)) < \rho$$

– a significant number of clauses that are not satisfied, i.e., there are many errors
– error correcting codes help, we guess,
– yes, we can do that

Constraint satisfaction problem, CSP

Definition

Let q be a natural number. A qCSP instance ϕ is a collection of functions $\phi_1, \phi_2, \cdots \phi_m$ from $\{0,1\}^n \to \{0,1\}$ such that each ϕ_i contains only q variables.
– $\mathrm{val}(\phi)$ is the maximum fraction of functions $\phi_1, \phi_2, \cdots, \phi_m$ that are satisfied by any assignment of the variables $x_1, x_2, \cdots, x_n$.
– q is called the arity of ϕ.

Gap CSP

Definition

For $q \in \mathbb{N}$, $\rho \leq 1$, we define the ρ-GAP qCSP to be the problem of the following form:
Given a qCSP instance ϕ,
If ϕ is satisfied, $\mathrm{val}(\phi) = 1$.
If ϕ is unsatisfied, then

$$\mathrm{val}(\phi) < \rho.$$

Hardness of ρ-GAP qCSP

For every $L \in \mathrm{NP}$, there is a polynomial time reduction R, for any x,

$$x \in L \Rightarrow \mathrm{val}(R(x)) = 1$$

$$x \notin L \Rightarrow \mathrm{val}(R(x)) < \rho$$

Hardness of GAP CSP

Theorem

There exist q, $\rho < 1$ such that ρ-GAP q CSP is NP-hard.

PCP verifier implies GAP CSP hard

Let $\mathrm{NP} \subseteq \mathrm{PCP}(\log n, 1)$. Suppose that V is a $(c \log n, q)$-PCP verifier for 3SAT. Then:

1) If x is in 3SAT, $\exists \pi$,

$$\Pr_{r \in_{\mathrm{R}} \{0,1\}^{c \log n}} [V^{\pi}(x, r)] = 1$$

2) If x is not in 3SAT, then for any π,

$$\Pr_{r \in_{\mathrm{R}} \{0,1\}^{c \log n}} [V^{\pi}(x, r)] \leq \frac{1}{2}$$

V queries q bits of π.

Here $V^{\pi}(x, r)$ uses q queries for π, $\pi_{i_1}, \cdots, \pi_{i_q}$ say.
Therefore, $V^{\pi}(x, r)$ is a function of q variables.
Let ϕ be the collection of $V^{\pi}(x, r)$ for all $r \in \{0, 1\}^{c \log n}$. Then ϕ is a qCSP instance.
The PCP verifier ensures that $\frac{1}{2}$-GAP qCSP is NP-hard.

ρ-GAPqCSP hardness implies $\mathrm{NP} \subseteq \mathrm{PCP}(\log n, 1)$

Suppose that there exist q, $\rho < 1$ such that ρ-GAPqCSP is NP-hard.
For every language $L \in \mathrm{NP}$, there is a reduction R such that for all x, $R(x)$ is a qCSP instance satisfying:

$$x \in L \Rightarrow \mathrm{val}(R(x)) = 1$$

$$x \notin L \Rightarrow \mathrm{val}(R(x)) < \rho$$

Then the PCP verifier V proceeds as follows: Let $R(x) = \{\phi_1, \phi_2, \cdots, \phi_m\}$

1. randomly pick ϕ_i
2. Query π for the variables of ϕ_i
3. Accept, if ϕ_i is satisfied.

Proof

Completeness: If $x \in L$, then $R(x)$ is satisfied, there is a proof π such that $\Pr[V^\pi(R(x)) = 1] = 1$.
Soundness: If $x \in L$, for any π,

$$\Pr[V^\pi(R(x)) = 1] < \rho.$$

By repeating several times,
for any π,

$$\Pr[V^\pi(R(x)) = 1] \leq \frac{1}{2}.$$

Hece

$$L \in \mathrm{PCP}(\log n, 1).$$

Hardness of GAP qCSP $\equiv$ Gap reduction

Any function $\psi : \{0,1\}^q \to \{0,1\}$ can be transformed into a set of 3CNF clauses.

A weak PCP verifier

Theorem

$$\mathrm{NP} \subseteq \mathrm{PCP}(\mathrm{poly}(n), 1).$$

Linearity test establishes the theorem.

Walsh Hadamard code- recall

In $GF(2)$,

$$\mathrm{WH}: \{0,1\}^* \to \{0,1\}^*$$

$$u \in \{0,1\}^n \mapsto \langle u \odot x \rangle_{x \in \{0,1\}^n}.$$

WH is regarded as a function from $\{0,1\}^n$ to $\{0,1\}$ or a string of length 2^{2^n}.

Random Subsum Principle

Theorem

If $u \neq v$, then

$$\Pr[u \odot x \neq v \odot x] = \frac{1}{2}.$$

Let $u_1 = 1$, $v_1 = 0$, for any x,

$$u \odot x = v \odot x$$

$$\Longleftrightarrow$$

$$x_1 = (v_2 - u_2)x_2 + \cdots (v_n - u_n)x_n$$

The only bit that is fixed is x_1. The number of such x's is $2^n/2$.

ECC

WH is an ECC of distance $\frac{1}{2}$.

WH is linear

Given a function $f: \{0,1\}^n \to \{0,1\}$, f is linear if and only if f is a WH codeword.
WH codeword

$$f(x_1, x_2, \cdots, x_n) = a_1x_1 + a_2x_2 + \cdots + a_nx_n.$$

In GF(2).

$$f \text{ is linear}$$

$$\Longleftrightarrow$$

$\forall x, y$

$$f(x+y) = f(x) + f(y).$$

Linearity test

Given a function $f: \{0,1\}^n \to \{0,1\}$, the algorithm tests whether f is linear or fare from any linear function.

$\mathcal{T}$:

1) Pick $x, y \in_{\mathrm{R}} \{0,1\}^n$,
2) Accepts if $f(x+y) = f(x) + f(y)$, and reject otherwise.

The test queues f only three bits.

Definition

Definition

Let $0 \leq \delta \leq 1$, $f, g : \{0,1\}^n \to \{0,1\}$ be functions.
We say that f, g are δ-close, if

$$\Pr_{x \in_R \{0,1\}^n}[f(x) = g(x)] \geq \delta.$$

Theorem

Completeness: If f is linear, then $\mathcal{T}$ accepts with prob 1.

Theorem

If $\mathcal{T}$ accepts with probability $\frac{1}{2} + \rho$, then f is 2ρ-close to some WH codeword.

Local decoder for WH: Recall

Given $f : \{0,1\}^n \to \{0,1\}$, suppose that f is $(1-\delta)$-close to a linear function $\widehat{f}$, for some $\delta < \frac{1}{4}$. Then $\widehat{f}$ is locally decoded.
We compute $\widehat{f}$ by
$\mathcal{T}$: for an input $x \in \{0,1\}^n$,

1) Pick $y \in_{\mathrm{R}} \{0,1\}^n$
2) Output $b = f(y) + f(y+x)$
   All in GF(2).

Result: With prob at least $1-2\delta$, $b = \widehat{f}(x)$.

Proof of the weak PCP for NP

QUADEEQ
Quadratic equations.
In GF(2)

$$u_1u_2 + u_3u_4 + u_1u_5 = 1$$

$$u_2u_3 + u_1u_4 = 0$$

A QUADEQ instance is a system of m equations. each is a quadratic over variables $u_1, u_2, \cdots u_n$, in GF(2).
The system is expressed by a matrix A of $m \times n^2$ and an m-dimensional vector b.
Solving the system is to find:
1) n^2-dimensional vector U such that

$$AU = b$$

2) U is the tensor product $u \otimes u$ for some n-dimensional vector u.

PCP verifier for QUADEQ

Given a QEADEQ instance $AU = b$, we check is there an assignment u such that
i) $U = u \otimes u$,
ii) $AU = b$,
The key is we are allowed to query only a constant bits from a proof π.
This is possible if we use the WH codewords.
Given $f : \{0,1\}^n \to$ to be expected the WH code of an assignment u
Given a function $g : \{0,1\}^{n^2} \to \{0,1\}$ to be expected the WH code of $u \otimes u$

PCP verifier V

$\mathcal{T}$:

1. Check whether or not f, g are linear.
   Suppose yes, and $f = \widehat{f}$, $g = \widehat{g}$.
2. Check $\widehat{g}$ is the codeword of $u \otimes u$.
3. Check if g encodes a satisfying assignment, by *random subsum principle*.

Hilbert space

Transfer GF(2^n) to $\{\pm 1\}^n$

- $b \mapsto (-1)^b$
- $0 \mapsto 1$
- $1 \mapsto -1$
- $\mathrm{XOR} \mapsto \cdot$

The Hilbert space consists of the functions from $\{\pm 1\}^n$ to $\mathbb{R}$ with

(i) $(f+g)(x) = f(x) + g(x)$

(ii) $(\alpha f)(x) = \alpha f(x)$

(iii) $\langle f, g \rangle = E[f(x)g(x)]$, the inner product, x is chosen uniformly and randomly.

The Fourier basis

For every $\alpha \subseteq [n]$,

$$\chi_\alpha(x) = \prod_{i \in \alpha} x_i$$

$$\chi_\emptyset = 1.$$

Theorem

1) The Fourier basis is an orthonormal basis
2) This is equivalent to Walsh-Hadamard codes

Given a linear function f:

$$f(x) = a_1 x_1 + a_2 x_2 + \cdots a_n x_n$$

Set $\alpha = \{i \mid a_i = 1\}$
Then
χ_α corresponds to f.

$\alpha = \beta$

For $\alpha, \beta \subseteq [n]$, define

$$\delta_{\alpha,\beta} = \langle \chi_\alpha, \chi_\beta \rangle.$$

For $\alpha = \beta$,

$$\begin{aligned} \delta_{\alpha,\beta} &= \langle \chi_\alpha, \chi_\beta \rangle \\ &= E_{x \in_{\mathrm{R}} \{\pm 1\}^n} [\chi_\alpha(x) \chi_\alpha(x)] \\ &= E_{x \in_{\mathrm{R}} \{\pm 1\}^n} [\prod_{i \in \alpha} x_i \prod_{i \in \alpha} x_i] = 1. \end{aligned} \tag{1}$$

$\alpha \neq \beta$

$$
\begin{aligned}
\delta_{\alpha,\beta} &= \langle \chi_\alpha, \chi_\beta \rangle \\
&= E_{x \in_R \{\pm 1\}^n}[\chi_\alpha(x)\chi_\beta(x)] \\
&= E_{x \in_R \{\pm 1\}^n}[\prod_{i \in \alpha} x_i \prod_{i \in \beta} x_i] \\
&= E_{x \in_R \{\pm 1\}^n}[\prod_{i \in \alpha \setminus \beta, \text{ or } i \in \beta \setminus \alpha} x_i] = 0.
\end{aligned} \tag{2}
$$

χ_α are orthonormal basis of the Hilbert space.

Fourier representation

$$\chi_\alpha(x) = \prod_{i \in \alpha} x_i$$

– a multi-linear function
Every function $f : \{\pm 1\}^n \to \mathbb{R}$,

$$f = \sum_{\alpha \subseteq [n]} \widehat{f}_\alpha \chi_\alpha,$$

$\widehat{f}_\alpha$ is the Fourier coefficients of f.

Parseval's identity

Lemma

For every $f, g : \{\pm 1\}^n \to \mathbb{R}$*,*

1) $\langle f, g \rangle = \sum_\alpha \widehat{f}_\alpha \widehat{g}_\alpha,$

2) *(Parseval's identity)* $\langle f, f \rangle = \sum_\alpha \widehat{f}_\alpha^2.$

$$\begin{aligned}
\langle f, g \rangle &= \langle \sum_\alpha \widehat{f}_\alpha \chi_\alpha, \sum_\beta \widehat{g}_\beta \chi_\beta \rangle \\
&= \sum_{\alpha,\beta} \widehat{f}_\alpha \widehat{g}_\beta \langle \chi_\alpha, \chi_\beta \rangle \\
&= \sum_{\alpha,\beta} \widehat{f}_\alpha \widehat{g}_\beta \delta_{\alpha,\beta} \\
&= \sum_\alpha \widehat{f}_\alpha \widehat{g}_\alpha. \qquad (3)
\end{aligned}$$

Boolean functions

$f : \{\pm 1\}^n \to \mathbb{R}$ is Boolean, if for every $x \in \{\pm 1\}^n$, $f(x) \in \{\pm 1\}$.
For Boolean function $f : \{\pm 1\}^n \to \{\pm 1\}$,

$$\langle f, f \rangle = E_x[f^2(x)] = 1.$$

χ_α

For $x, y \in \{\pm 1\}^n$,
define

$$xy = (x_1y_1, x_2y_2, \cdots, x_ny_n).$$

$$\begin{aligned}\chi_\alpha(xy) &= \prod_{i\in\alpha}(xy)_i \\ &= \prod_{i\in\alpha} x_i \prod_{i\in\alpha} y_i \\ &= \chi_\alpha(x)\chi_\alpha(y). \end{aligned} \tag{4}$$

Inner product of Boolean functions

For Boolean functions f, g,

$$\begin{aligned}\langle f, g\rangle &= E_x[f(x)g)x)] \\ &= \text{the fraction of } x \text{ at which } f(x) = g(x) \\ &- \text{the fraction of } x \text{ at which } f(x) \neq g(x).\end{aligned} \tag{5}$$

If $\langle f, g\rangle = \epsilon$, then

$$\Pr[f(x) = g(x)] = \frac{1}{2} + \frac{\epsilon}{2}.$$

Therefore, of there is an α such that $\widehat{f}_\alpha = \langle f, \chi_\alpha\rangle = \epsilon$, then f is $(\frac{1}{2} + \frac{\epsilon}{2})$-close to the linear function χ_α.

Linearity test

Theorem

Suppose that $f : \{\pm 1\}^n \to \{\pm 1\}$ *satisfies*

$$\Pr_{x,y \in_R \{\pm 1\}^n}[f(xy) = f(x)f(y)] \geq \frac{1}{2} + \epsilon.$$

Then there is an α *such that*

$$\widehat{f_\alpha} \geq 2 \cdot \epsilon.$$

Intuition

Intuition If the linearity test accepts f with a probability slightly better than $\frac{1}{2}$, then there is a linear function χ_α that is close to f. This means that, if f is far from any linear function, then the test accepts f with probability no larger than $\frac{1}{2}$.
Due to

$$\langle f, f \rangle = \sum_\alpha \widehat{f}_\alpha^2 = 1$$

with probability $\widehat{f}_\alpha^2$ to choose α, decodes the linear function χ_α that is close to f, if any.

Proof - I

Assume

$$\Pr[f(xy) = f(x)f(y)] \geq \frac{1}{2} + \epsilon.$$

Then

$$E_{e,y}[f(xy)f(x)f(y)] \geq \frac{1}{2} - (\frac{1}{2} - \epsilon) = 2\epsilon.$$

$$\begin{aligned} 2\epsilon &\leq E_{x,y}[f(xy)f(x)f(y)] \\ &= E_{x,y}[\sum_{\alpha} \widehat{f}_{\alpha}\chi_{\alpha}(xy) \sum_{\beta} \widehat{f}_{\beta}\chi_{\beta}(x) \sum_{\gamma} \widehat{f}_{\gamma}\chi_{\gamma}(y)] \\ &= E_{x,y}[\sum_{\alpha} \widehat{f}_{\alpha}\chi_{\alpha}(x)\chi_{\alpha}(y) \sum_{\beta} \widehat{f}_{\beta}\chi_{\beta}(x) \sum_{\gamma} \widehat{f}_{\gamma}\chi_{\gamma}(y)] \end{aligned}$$

Proof - II

$$= E_{x,y}[\sum_{\alpha,\beta,\gamma} \widehat{f_\alpha}\widehat{f_\beta}\widehat{f_\gamma}\chi_\alpha(x)\chi_\alpha(y)\chi_\beta(x)\chi_\gamma(y)]$$

$$= \sum_{\alpha,\beta,\gamma} \widehat{f_\alpha}\widehat{f_\beta}\widehat{f_\gamma}E_x[\chi_\alpha(x)\chi_\beta(y)]E_y[\chi_\alpha(y)\chi_\gamma(y)]$$

$$(x, y \text{ are independent})$$

$$= \sum_{\alpha,\beta,\gamma} \widehat{f_\alpha}\widehat{f_\beta}\widehat{f_\gamma}\delta_{\alpha,\beta}\delta_{\alpha,\gamma}$$

$$= \sum_\alpha \widehat{f_\alpha^3} \le (\max_\alpha \widehat{f_\alpha}) \sum_\alpha \widehat{f_\alpha^2} = \max_\alpha \widehat{f_\alpha}. \tag{6}$$

General idea of Fourier analysis

There are many applications.
If a function is correlated with itself in some structured way, then it belongs to a small set of functions.

Exercises

(1) Give a probabilistic polynomial time algorithm that given a 3CNF formula ϕ with exactly three distinct variables in each clause, outputs an assignment satisfying at least a $\frac{7}{8}$ fraction of ϕ's clauses.

(2) Give a deterministic polynomial time algorithm with the same approximation guarantee as Exercise 1 above.

(3) Show a polynomial time algorithm that given a satisfiable 2CSP instance ϕ over binary alphabet with m clauses outputs a satisfying assignment for ϕ.

(4) Show a deterministic poly $(n, 2^q)$-time algorithm that given a qCSP-instance ϕ over binary alphabet with m clauses outputs an assignment satisfying $m/2^q$ of the constraints of ϕ.

Exercises

(5) Suppose that $G = (V, E)$ is an (n, d, λ)-expander. Show that for any $S \subset V$ of size $\leq \frac{n}{2}$, the following holds:

$$\Pr_{(u,v)\in_{\mathrm{R}} E}[u \in S \wedge v \in S] \leq \frac{|S|}{n}(\frac{1}{2} + \frac{\lambda}{2}).$$