6.S897 Algebra and Computation May 2, 2012

Probababilistically Checkable Proofs

Instructor: Madhu Sudan *Scribe: Jin Hao Wan*

1 Overview

Today we talk about probabilistically checkable proofs (PCP) and how algebra helps in constructing such proofs. First, we revisit and formally simplify the conventional notion of a theorem and a proof. Next, we introduce the definition of a PCP. Finally, we discuss how to use algebra to construct PCPs for the graph 3-colorability problem.

2 Theorems & Proofs

What is a theorem and what is a proof? From a "naive" point of view, each is just a string of bits. This simple notion holds even for abstract proofs from geometry or topology: when a geometer or a topologist publishes her proof, she need to write it out as a string of bits! More formally, given integers l and n, where $l \leq n$, a theorem T is an element of $\{0,1\}^l$; T is true if and only if there exists a proof $\pi \in \{0,1\}^n$ that "proves" it.

Note that in the above notion we do not worry about how n and l are related: n is simply given as part of the theorem. This way, we avoid a decidability mess; one can simply check all 2^n possible proofs to decide if a given theorem is true.

In order to decide whether a proof proves a theorem, we need to have a system of logic. This is captured by a *verifier*:

Definition 2.1 (Verifier)**.** A verifier V is a polynomial time (in n) Turing Machine mapping $\{0,1\}^l \times \{0,1\}^n$ to $\{0,1\}$ such that it satisfies the following two conditions:

- *Completeness*: If $T \in \{0,1\}^l$ is true, $\exists \pi \in \{0,1\}^n$ such that $V(T,\pi) = 1$.
- *Soundness*: If $T \in \{0,1\}^l$ is false, $V(T,\pi) = 0$ for all $\pi \in \{0,1\}^n$.

We can associate theorems to any decidability problem in a straight-forward way. For example, given three integers n, u, and l, we would like to decide if there exists a factor d of n such that $l \leq d \leq u$. A theorem for this decidability problem can be specified by the triple $T = (n,u,l)$. To prove T, we only need to find an integer d of bounded size such that $d|n$ and $l \leq d \leq u$. It can be shown that this underlying logic system is *complete*, i.e., any proof system can be *reduced* to it. The concept of completeness and reduction is analogous to the one in complexity theory; in particular, reduction is captured by the following definition of *equivalence*:

Definition 2.2 (Equivalence of Proof Systems)**.** Two proof systems, V and V', are equivalent if every theorem T in V with proof of length n can be reduced in polynomial time (in n) to a theorem T' (of length *poly*(n)) in V' such that T is true iff T' is true. Moreover, the length of T''s proof must be *poly*(n).

Note that V could be a logic system for theorems from combinatorics while V' could be a logic system for theorems from number theory. Once we show that V and V' are equivalent, we can check every theorem in one system with the verifier of the other. This can give us an edge if verifying theorems in the other system is easier. For example, from complexity theory, we have the following fact:

Theorem 2.3 (Karp's Theorem)**.** *Graph* 3*-coloring is* NP*-complete.*

Therefore, we can reduce any proof system based a NP problem to the proof system based the 3-colorability problem. That is, given a theorem T from a logic system, we can somehow transform it into a graph G of size $poly(n)$, where $n = |T|$; then T is true if and only if G is 3-colorable. Under the new system, the proof is a proposed 3-coloring of G. In most cases, this proof is more convenient to verify since, if it is wrong, it has an easy *localizable* error (two adjacent vertices of the same color). However, finding such an error deterministically still requires scanning over the entire G. As we will see, PCP tries to make this verification task even easier.

3 Definition of PCP

In order to simplify the verification task, we can make the verifier more powerful by allowing it to toss coins. Such verifier is called a probabilistically checkable proof (PCP) system.

Definition 3.1 (Probabilistically Checkable Proof System)**.** Given an alphabet Σ, a probabilistically checkable proof system, V_{PCP}, is a randomized verifier that, on input $T \in \Sigma^l$ and $\pi \in \Sigma^n$, generates a random string R of length r and makes at most q queries to π (based on T and R) such that it satisfies the following conditions:

- *Completeness*: If T is true, $\exists \pi$ such that $\Pr_R[V^{\pi}_{PCP}(T, \pi) = 1] = 1$.
- *Soundness*: If T is false, $\forall \pi$ we have that $\Pr_R[V^{\pi}_{PCP}(T, \pi) = 1] \leq \frac{1}{2}$.

Two comments about this definition:

1. The way V_{PCP} queries a proof π is based entirely on the theorem T and the random string R. In other words, V_{PCP} is not adaptive. This restriction can be relaxed.
2. The probabilities in the two conditions (1 and $\frac{1}{2}$) can be parametrized.
3. The best PCP system (for the 3-colorability problem) only need to make at most 3 queries (at the expense of using more randomness).

As we can see, one can work with a variety of PCP systems by changing some requirements/parameters. For this lecture, however, we are only working with the above defition. Furthermore, we restrict our system to the problem of 3-colorability. Our goal is to construct a PCP system with $O(poly(\log n))$ queries.

4 Overview of Proof

Given a graph G whose vertex set V has size n, we are going to represent G as a function $E : V \times V \mapsto \{0,1\}$ and a proposed 3-coloring as a function $A : V \mapsto \{-1,0,1\}$. Therefore, A is a valid 3-coloring for G if and only if $\forall\ i,j$ either $E(i,j) = 0$ or $A(i) \neq A(j)$.

We bring algebra into the picture by embedding V injectively in a finite field $\mathbb{F}$ ($|F| = O(n)$). We will use the following related facts about polynomial,

Theorem 4.1. *(Two Facts about Polynomials)*
Given a subset $S \subseteq \mathbb{F}$ and a univariate polynomail $f : S \mapsto \mathbb{F}$, $\exists$ a univariate polynomial $\hat{f} : \mathbb{F} \mapsto \mathbb{F}$ of degree $\leq |S|$ such that

$$\hat{f}(\alpha) = f(\alpha) \quad \forall \alpha \in S.$$

Given a subset $S \subseteq \mathbb{F}$ and a bivariate polynomial $f : S \times S \mapsto \mathbb{F}$, $\exists$ a bivariate polynomial $\hat{f} : \mathbb{F} \times \mathbb{F} \mapsto \mathbb{F}$ of degree $\leq |S|$ in each variable such that

$$\hat{f}(\alpha,\beta) = f(\alpha,\beta) \quad \forall (\alpha,\beta) \in S \times S.$$

Based on these two facts, we can "interpolate" A and E into two new polynomials $\hat{A}$ and $\hat{E}$, respectively, such that $deg(\hat{A}) \leq n$, $deg_x(\hat{E}) \leq n$ and $deg_y(\hat{E}) \leq n$. Note that since $\hat{A}$ and $\hat{E}$ agree with A and E on V completely, we can use them instead to verify 3-colorability. So, to recapture the requirements we discussed about a valid 3-coloring, a graph represented by $\hat{E} : \mathbb{F} \times \mathbb{F} \mapsto \mathbb{F}$, where $deg_{x,y}(\hat{E}) \leq n$, is 3-colorable if and only if $\exists \hat{A} : \mathbb{F} \mapsto \mathbb{F}$, where $deg(\hat{A}) \leq n$, such that,

1. $\hat{A}$ uses no more than three colors: $\forall \alpha \in V$, $(\hat{A}(\alpha)+1)(\hat{A}(\alpha))(\hat{A}(\alpha)-1) = 0$.
2. $\hat{A}$ is a valid 3-coloring: $\forall\ \alpha,\beta \in V$, $\hat{E}(\alpha,\beta)\prod_{i\in\{-2,-1,1,2\}}(\hat{A}(\alpha) - \hat{A}(\beta) - i) = 0$.

Note that $\{-2,-1,1,2\}$ contains all possible differences between two distinct elements of $\{-1,0,1\}$.

Therefore, given a proof $\hat{A}$, the verifier V_{PCP} need to make two *sequential* checks:

(a) Syntactic check: is the degree of $\hat{A} \leq n$?

(b) Semantic check: does $\hat{A}$ satisfy conditions 1 and 2 above?

Intuitively, the syntactic check verifies if the proof is in the right "language;" it is the more technical part of this lecture. For now, we will assume that V_{PCP} knows how to implement syntactic check. More precisely, we are making the following assumptions:

1. Given a univariate polynomial $A : \mathbb{F} \mapsto \mathbb{F}$, represented by an array containing its evaluations at all $|\mathbb{F}|$ points, V_{PCP} can verify that $deg(A) \leq d$.
2. Similarly, given a bivariate polynomial $B : \mathbb{F} \times \mathbb{F} \mapsto \mathbb{F}$, represented by a matrix containing its evaluations at all $|\mathbb{F}|^2$ points, V_{PCP} can verify that $deg_x(B) \leq d$ and $deg_y(B) \leq d$.

We will see that the actual proof (whose exact form we will specify in the next section) contains not only $\hat{A}$, but some related univariate and bivariate polynomials as well; therefore, the second assumption is required. For now we can assume that our proof passes the syntactic check and move on to implementing the semantic check.

5 Semantic Check

6 Syntactic Check