Fall 2014 Randomized Algorithms Oct 22, 2014

Lecture 5

Prof. Friedrich Eisenbrand *Scribes: Manuel Francesco Aprile*

In this lecture we introduce the notion of polynomial identity testing and discuss an application to the problem of finding a matching of maximum cardinality in a graph.

Polynomial Identity Testing

Problem

Let $\mathbb{F}$ be a field (for instance $\mathbb{F} = \mathbb{Q}, \mathbb{Z}_p$ with p prime, $\dots$) and $f(x_1, \dots, x_n)$, $g(x_1, \dots, x_n) \in \mathbb{F}[x_1, \dots, x_n]$ two polynomials in n variables and coefficients in $\mathbb{F}$. Our goal is to decide whether $f \equiv g$, i.e. f and g are identical. More precisely, if $f = \sum_{\alpha \in \mathbb{N}_0^n} f_\alpha x^\alpha$, $g = \sum_{\alpha \in \mathbb{N}_0^n} g_\alpha x^\alpha$, we say that $f \equiv g$ if and only if $f_\alpha = g_\alpha$ for each $\alpha \in \mathbb{N}_0^n$. Note that $f \equiv g$ if and only if $f - g \equiv 0$.

Motivation

In many cases checking identity using the definition given above is not efficient, because we are not given the extended representation of the f, g. For instance, consider two square matrices $A, B \in \mathbb{F}[x_1, \dots, x_n]^{d \times d}$, and suppose we want to know whether their determinants (that are polynomial in $\mathbb{F}[x_1, \dots, x_n]$) are the same, i.e., $\det(A) \equiv \det(B)$. The standard formula for the determinant of a matrix is as follows:

$$\det(A) = \sum_{\pi \in S_d} \operatorname{sign}(\pi) \prod_{i=1}^{d} a_{i,\pi(i)}. \tag{1}$$

Applying this formula is not efficient as there are exponentially many permutations to consider ($|S^d| = d!$). Therefore we look for an efficient algorithm that decides if two polynomials are identical (or, as pointed out before, if a polynomial is identically zero) without actually compute any extended representation. The idea is to evaluate the polynomials in some random points: if the results are different, than we know for sure that the polynomials are different. Otherwise, we decide that the polynomials are identical. The following lemma bounds the probability that we make a mistake in this decision.

Lemma 1 (Schwartz-Zippel Lemma) *Let $\mathbb{F}$ be a field and $f(x_1, \dots, x_n) \in \mathbb{F}[x_1, \dots, x_n]$ of total degree d. Let $S \subset \mathbb{F}$ a finite set, and suppose we sample $(x_1^*, \dots, x_n^*)$ from S uniformly at random. If $f \not\equiv 0$, then*

$$\mathbb{P}(f(x_1^*, \dots, x_n^*) = 0) \leq \frac{d}{|S|}$$

Proof The proof is by induction on n. If $n = 1$, $f(x_1)$ has at most d roots (since $\mathbb{F}$ is a field). Therefore $\mathbb{P}(f(x_1^*) = 0) \leq \frac{d}{|S|}$.

Suppose $n > 1$. We write f as a polynomial in x_n, with coefficients in $\mathbb{F}[x_1, \ldots, x_{n-1}]$:

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{k} g_i(x_1, \ldots, x_{n-1}) x_n^i$$

with $k \leq d, g_k \not\equiv 0$. Let E_1 be the event that $f(x_1^*, \ldots, x_n^*) = 0$, and E_2 be the event that $g_k(x_1^*, \ldots, x_{n-1}^*) = 0$. Note that since g_k has degree at most $d - k$, by induction $\mathbb{P}(E_1) \leq \frac{d-k}{|S|}$. Now:

$$\mathbb{P}(E_1) = \mathbb{P}(E_1 \wedge E_2) + \mathbb{P}(E_1 \wedge \neg E_2) \leq \mathbb{P}(E_2) + \mathbb{P}(E_1 \wedge \neg E_2) \leq$$

$$\frac{d-k}{|S|} + \mathbb{P}(E_1 | \neg E_2) \cdot \mathbb{P}(\neg E_2) \leq \frac{d-k}{|S|} + \mathbb{P}(E_1 | \neg E_2) \leq \frac{d-k}{|S|} + \frac{k}{|S|} = \frac{d}{|S|}$$

where in the last inequality we have used the fact that, if $\neg E_2$ holds, the coefficient of x_n^k is not zero, hence f is a polynomial of degree k in x_n. ■

Now suppose we want to know if a polynomial f is identically zero, and we take S such that $|S| \geq 2\deg(f)$. Thanks to lemma 1, if $f \not\equiv 0$, the probability that a random point from S is a root of f is less thank $1/2$. We can exponentially boost this probability by making many independent samples.

Application to Max Cardinality Matching

Problem

Let $G(V, E)$ be a graph. A matching M of G is a set of disjoint edges, i.e., $M \subset E$ such that $e_1 \cap e_2 = \emptyset$ for any distinct $e_1, e_2 \in M$. A matching is said to be perfect if each vertex of G is an endpoint of an edge in M ($\bigcup_{e \in M} e = V$). The goal is to find a matching with maximum cardinality (therefore, to determine if the graph has a perfect matching). This problem can be solved in polynomial time (the first algorithm for this is Edmonds' Blossom-Shrink algorithm, [1]).

Tutte Matrix

To relate this problem to polynomial identity testing, we first introduce a notion that will be helpful in determining whether G has a perfect matching or not. What follows is from [2].

Definition 2 *Given a graph $G(V, E)$ with n vertices, its Tutte Matrix A_G is an $n \times n$ matrix with entries*

$$A_{ij} = \begin{cases} x_{ij} & \text{if } (i,j) \in E \text{ and } i \leq j \\ -x_{ij} & \text{if } (i,j) \in E \text{ and } i \geq j \\ 0 & \text{otherwise} \end{cases}$$

Lemma 3 *Let $G(V,E)$ be a graph, then* $\det(A_G) = 0 \iff$ *G has no perfect matching.*

Before we prove the lemma, let us see why it is important for our goal. Suppose that $|V| = n$, $|E| = m$. Then $\det(A_G)$ is a polynomial in m variables, coefficients in $\mathbb{Q}$ and it has degree at most n. Suppose that it is not identically zero. So if we choose $S = \{1, \ldots, 2n\}^m$, sample $(x_e, \forall e \in E)$ uniformly at random from S and replace the values in A_G, the probability that $\det(A_G) = 0$ is at most $1/2$. By iterating i times, we can state whether G has a perfect matching or not with a probability of error of at most $1/2^i$.

Proof Suppose first that G has a perfect matching M, we have to show that $\det(A_G) \neq 0$. To do this, we will find a point in $\mathbb{Q}^m$ where the determinant is non-zero. Without loss of generality we may assume that $M = \{(1,2), (2,3), \ldots, (n-1,n)\}$. Now we choose $x^* = (x^*_e, \forall e \in E)$ such that $x^*_{i,i+1} = 1$, and all the other components are 0. Then

$$A_G(x^*) = \begin{pmatrix} 0 & 1 & 0 & 0 & \ldots \\ -1 & 0 & 0 & 0 & \ldots \\ 0 & 0 & 0 & 1 & \ldots \\ 0 & 0 & -1 & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

i.e., $A_G(x^*)$ is a block matrix with determinant 1.

Now suppose that $\det(A_G) \neq 0$, we want to show that there is a perfect matching. From formula 1 we know that $\det(A_G)$ is a sum of monomials, one for each permutation $\pi \in S_n$. First note that for a monomial $\prod_{i=1}^n A_{i,\pi(i)}$ to be non-zero we must have $(i, \pi(i)) \in E \; \forall\, i$. If π satisfies this property, we say that π is supported by G and we denote by S^*_n the set of permutations that are supported by G. What has been said implies that the sum in formula 1 can be restricted to $\pi \in S^*_n$. Moreover, each permutation π decomposes into disjoint cycles. Note that if π is supported by G and all cycles in π are even, than we can obtain a perfect matching just picking alternating edges in these cycles. Therefore we only have to show that such a π exists. We can write:

$$\det(A_G) = \sum_{\substack{\pi \in S^*_n, \\ \pi \text{ has an odd cycle}}} \operatorname{sign}(\pi) \prod_{i=1}^n a_{i,\pi(i)} + \sum_{\substack{\pi \in S^*_n, \\ \pi \text{ has no odd cycles}}} \operatorname{sign}(\pi) \prod_{i=1}^n a_{i,\pi(i)}$$

Since $\det(A_G) \neq 0$, if we show that the first summand is 0 we are done, because then there must be a π supported by G which has only even cycles. The idea is to pair each permutation π in the first sum with a "friend" $f(\pi)$ such that $f(\pi)$ can be decomposed in the same cycles of π, but the odd cycle which contains the vertex of higher index has reversed order. Since π has at least one odd cycle (hence, at least one cycle with 3 or more elements), $f(\pi)$

and π are different, therefore the set of all permutations in S_n^* with at least one odd cycle can be partitioned in such pairs. But $f(\pi)$ and π are such that they have opposite sign and $\prod_{i=1}^n a_{i,\pi(i)} = \prod_{i=1}^n a_{i,f(\pi(i))}$, hence they cancel out, and the whole first summand is zero.

■

As said before, thanks to lemma 3 we can tell whether G has a perfect matching by computing the determinant of A_G, which takes $O(n^\omega)$ time, where n is the number of vertices of G and ω is the exponent needed to perform a matrix multiplication. If the answer is positive, the matching can be found with the following procedure: start with an empty matching M, remove an edge e from G and check if $G \setminus e$ has a perfect matching. If it doesn't (hence any perfect matching of G must contain e) add e to M and remove from G e and its endpoints; then repeat until G is empty. The number of iterations needed is $O(|E|) = O(n^2)$, therefore the total running time of the algorithm is $O(n^{\omega+2})$. Note that, in case G has no perfect matching, we can find a matching that has maximum cardinality with high probability. Indeed, suppose we want to check if G has a matching of cardinality k. We add $n - 2k$ vertices connected to every vertex of G, then the new graph has a perfect matching if and only if G has a matching of cardinality k. In this way we find a maximal k such that G has a matching of cardinality k with binary search, which adds a logarithmic factor to the running time: therefore the algorithm takes $O(n^{\omega+2} \log n)$ time.

References

[1] Edmonds, Jack (1965). *Paths, trees, and flowers.* Canad. J. Math. 17: 449-467.

[2] W.T. Tutte (1947). *The factorization of linear graphs.* J. London Math. Soc. 22 (2): 107111.