Algorithmic Combinatorics - Lecture #28

1.2.2017

I. Polynomial Identity Testing

A. Review

1. Problem statement
   a) Q: Given a polynomial $h \in \mathbb{F}[x_1, \ldots, x_n]$, is $h = 0$?
2. Clarifying the problem.
   a) Def: A polynomial is a finite linear combination of monomials with coefficients in $\mathbb{F}$.
   b) A polynomial is the zero polynomial if all of its coefficients are equal to zero.
      (i) Remark: This is the algebraic definition of the zero polynomial.
      (ii) The computer-scientific definition would be if it is identically zero as a function: for all choice of input variables, the polynomial evaluates to zero.
      (iii) These are equivalent over fields of infinite characteristic, but not for finite fields, eg.
      $$f = x_1^2 + x_1 \quad \text{over } \mathbb{F}_2.$$
      (iv) Lead to different problems: we focus on the algebraic definition.

B. Oracle-access model.

1. How is our polynomial presented to us?
   a) If as in the definition — a linear combination of monomials — this problem is trivial

(i) Simply check each coefficient to see if any of them are non-zero.

b) (Un)fortunately, there are many ways to skin a cat, and ~~always~~ almost as many ways to be given a polynomial.

~~NB eg: could be given in factored form~~

(i) eg: polynomial could be a linear combination of products of factors; eg

$$f(x_1,x_2) = (x_1+x_2)^2 + (x_1-2x_2)(2x_1-x_2)$$
$$= 0 \text{ over } \mathbb{F}_3.$$

(ii) multiplying out can be exponentially large $\rightarrow$ not efficient.

2. Oracle access

a) We shall assume that we do not have access to the polynomial itself, but rather just the polynomial function

b) i.e. we can ask what value the polynomial takes at given points

C. Zero sets of polynomials

1. Algorithmic strategy.

a) Essentially the only thing we can do is test the polynomial at a number of different points, and see if we get any non-zero values.

b) How many zeroes do we need to see before we can be satisfied that the polynomial is indeed the zero polynomial?

2. One-dimensional setting

a) <u>Fact</u>: If $f \in \mathbb{F}[x]$ is a non-zero polynomial of degree at most d, then f can have at most d zeroes

b) Pf sketch:

(i) For every zero, have a linear factor of f.

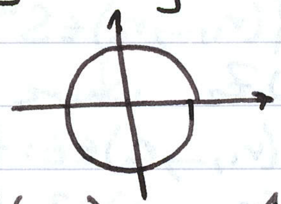
(ii) Induct on quotient (degree)

(iii) Remark: holds for polynomials over any <u>integral domain</u>: don't need a field.

3. Multi-dimensional setting

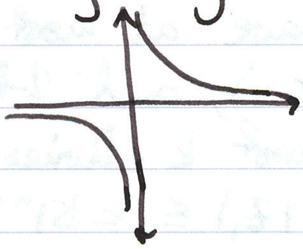
a) Zero sets of multivariate polynomials can be much larger

(i) <u>eg</u>: $f(x,y) = x^2+y^2-1 \in \mathbb{R}[x,y]$



infinitely many zeroes!

(ii) $f(x,y) = xy - 1 \in \mathbb{R}[x,y]$



4. Schwartz-Zippel Lemma

a) Idea: zero sets are structured: cannot intersect a cube too often.

b) <u>Theorem</u> (The Schwartz-Zippel Lemma)

Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a non-zero polynomial of degree $d \geq 0$, and $S \subseteq \mathbb{F}$ a finite set. Then $|\{(r_1, \ldots, r_n) \in S^n : f(r_1, \ldots, r_n) = 0\}| \leq d|S|^{n-1}$.

c) Historical remarks

(i) Discovered for the purposes of polynomial identity testing

(ii) Similar results obtained independently by DeMillo-Lipton (1978), Zippel (1979), Schwartz (1980).

d) Proof of Schwartz-Zippel

(i) Induction on n.

(ii) Base case: $n=1$ $\equiv$ our earlier fact ✓

(iii) Induction step. Let x_n be a variable, and let $k \le d$ be the highest power of x_n appearing in f.

(iv) $\Rightarrow f(x_1, \dots, x_n) = \sum_{i=0}^{k} f_i(x_1, \dots, x_{n-1}) x_n^i$, where each $f_i \in \mathbb{F}[x_1, \dots, x_{n-1}]$.

(v) Let $Z = \{(r_1, \dots, r_n) \in S^n : f(r_1, \dots, r_n) = 0\}$.

(vi) $Z = Z_1 \cup Z_2$, where

$Z_1 = \{(r_1, \dots, r_n) \in Z : f_k(r_1, \dots, r_{n-1}) \ne 0\}$

$Z_2 = \{(r_1, \dots, r_n) \in Z : f_k(r_1, \dots, r_{n-1}) = 0\}$

(vii) In Z_1, have at most $|S|^{n-1}$ choices for $(r_1, \dots, r_{n-1})$, and 1-D case $\Rightarrow$ at most k choices for r_n

$\Rightarrow |Z_1| \le k|S|^{n-1}$.

(viii) f_k is a polynomial of degree $\le d-k$

I.H. $\Rightarrow$ $\le (d-k)|S|^{n-2}$ choices for $(r_1, \dots, r_{n-1})$

$\le |S|$ choices for r_n

$\Rightarrow |Z_2| \le (d-k)|S|^{n-1}$

(ix) $\Rightarrow |Z| \le d|S|^{n-1}$ ✓ $\square$.

D. Randomised algorithm

1. Testing random points

a) Schwartz-Zippel $\Rightarrow$ suffices to test a bounded (but large) number of points.

b) Using randomness gives huge efficiency boost.

2. Algorithm

a) ~~Fix [illegible]~~ Input: polynomial $h \in \mathbb{F}[x_1, \ldots, x_n]$ of degree $\leq d$. Q: is $h = 0$?

b) Fix $S \subseteq \mathbb{F}$, $|S| = 2d$, arbitrarily.

c) Choose $r_1, \ldots, r_n \in S^n$ uniformly at random.

d) If $h(r_1, \ldots, r_n) = 0$, return $h = 0$. Otherwise, return $h \neq 0$.

3. Analysis

a) No false negatives, only false positives.

b) Schwartz-Zippel
$$\Rightarrow \mathbb{P}(\text{false positive}) \leq \frac{d|S|^{n-1}}{|S|^n} = \frac{1}{2}.$$

c) Testing at k independent points in S^n
$$\Rightarrow \mathbb{P}(\text{false positive}) \leq \frac{1}{2^k}.$$

4. Small fields

a) What if $|\mathbb{F}| < 2d$, so we cannot choose S?

b) If $d \geq |\mathbb{F}|$, can make a non-zero polynomial that vanishes everywhere

(i) $\Rightarrow$ impossible to distinguish over $\mathbb{F}$

(ii) either restrict $d < |\mathbb{F}|$, or work over a field extension instead.

II. Bipartite Matchings

A. Motivation

1. Application of polynomial identity testing: determining if a bipartite graph has a perfect matching.

2. Already have the Augmenting Path Algorithm,

but that is a bit long and unwieldy.

3. This algorithm is simple, and more easily generalised to the non-bipartite setting.

B. Framework

1. Given bipartite graph $G=(U \sqcup V, E)$, $U=\{u_1, \ldots, u_n\}$, $V=\{v_1, \ldots, v_n\}$.
2. Can define a (bipartite) adjacency matrix $A=(a_{ij})_{1\le i,j \le n}$,
$$a_{ij} = \begin{cases} 1 & \text{if } \{u_i, v_j\} \in E \\ 0 & \text{otherwise.} \end{cases}$$

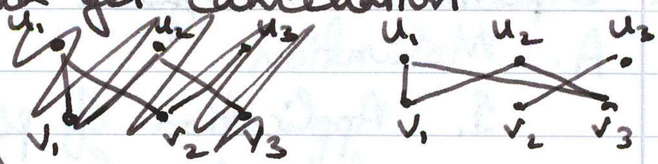
C. Permanents, Determinants and Matchings

1. Perfect matching in G
$$\Updownarrow$$
Permutation $\sigma \in S_n$ s.t.
$$\{\{u_1, v_{\sigma(1)}\}, \{u_2, v_{\sigma(2)}\}, \ldots, \{u_n, v_{\sigma(n)}\}\} \subseteq E(G)$$
$$\Updownarrow$$
Permutation $\sigma \in S_n$ s.t. $\prod_{i=1}^{n} a_{i\sigma(i)} = 1$

2. $\Rightarrow$ $\mathrm{per}(A) := \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)} = \#$ of perfect matchings in G.
   a) Problem: permanent is ~~NP-complete~~ NP-hard to compute!

3. Magically, introducing a sign factor gives the determinant, which is easy to compute!
$$\det(A) = \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) \prod_{i=1}^{n} a_{i\sigma(i)}$$
   a) Problem: can get cancellation

4. Example: $G =$



$$A = \begin{pmatrix} \boxed{1} & 0 & 1 \\ 1 & 0 & \boxed{1} \\ 0 & \boxed{1} & 0 \end{pmatrix}$$

$\mathrm{per}(A) = 2$

$\det(A) = 1 - 1 = 0.$

5. $\therefore \det(A) \neq 0 \Rightarrow \exists$ perfect matching, but $\det(A) = 0 \not\Rightarrow \nexists$ perfect matching.

D. Introducing variables

1. The fix

a) To prevent this harmful cancellation, we replace the entries of A with variables.

b) Let $\tilde{A} = (\tilde{a}_{ij})_{1 \le i,j \le n}$, where
$$\tilde{a}_{ij} = \begin{cases} x_{ij} & \text{if } \{u_i, v_j\} \in E \\ 0 & \text{otherwise.} \end{cases}$$

c) Now $\det(\tilde{A})$ is a polynomial in $\mathbb{F}[x_{11}, x_{12}, \ldots, x_{nn}]$ of degree at most n.

3. Characterisation

a) <u>Lemma</u>: $\det(\tilde{A}) \neq 0 \iff G$ has a perfect matching.

b) Proof

(i) $\Rightarrow$: If $\det(\tilde{A}) \neq 0$, there is some monomial with a non-zero coefficient.

monomial: $\prod_{i=1}^{n} x_{i\sigma(i)}$ for some $\sigma \in S_n$ defining a matching ✓

(ii) $\Leftarrow$: If we have a matching corresponding to $\sigma \in S_n$, let
$$x_{ij} = \begin{cases} 1 & \text{if } j = \sigma(i) \\ 0 & \text{otherwise} \end{cases}$$
$\Rightarrow \det(\tilde{A})(x) = \pm 1 \neq 0$ ✓ $\square$.

2. Example: $G =$ 

$$\tilde{A} = \begin{pmatrix} x_{11} & 0 & x_{13} \\ x_{21} & 0 & x_{23} \\ 0 & x_{32} & 0 \end{pmatrix}, \quad \det(\tilde{A}) = x_{21}x_{13}x_{32} - x_{11}x_{23}x_{32}.$$

E. The algorithm

1. Fix some set $S \subseteq \mathbb{F}$, $|S| = 2n$.
2. Construct $\tilde{A}$, substituting a uniformly random value from S for each edge variable x_{ij}
3. Compute $\det(\tilde{A})$ with these values.
4. If $\det(\tilde{A}) \neq 0$, then G has a perfect matching.
5. If $\det(\tilde{A}) = 0$, return: G has no perfect matching.

F. Analysis and closing remarks

1. Since $\det(\tilde{A})$ is a polynomial of degree $\leq n$, Schwartz-Zippel $\Rightarrow$ prob. of a false negative is at most $\frac{1}{2}$.
2. Which field?
   a) If $\mathbb{F} = \mathbb{R}$ and we take $S = [2n]$, we could deal with numbers as large as $(2n)^n$ $\rightarrow$ not good for computers
   b) Simpler to work over a finite field, eg. $\mathbb{F}_p$ for some prime $2n \leq p < 4n$ $\rightarrow$ bounded calculations.
3. Non-bipartite setting
   a) In the general case, an n-vertex graph has an $n \times n$ adjacency matrix, but only $\frac{n}{2}$ edges in a matching $\rightarrow$ no one-to-one correspondence b/w determinant monomials and matchings.
   b) Need to be a bit cleverer - see HW.
4. Any questions?