Learning Read-Constant Polynomials of Constant Degree Modulo Composites

Arkadev Chattopadhyay[1] Ricard Gavaldà[2]
Kristoffer Arnsfelt Hansen[3] Denis Thérien[4]

[1]University of Toronto

[2]Universitat Politècnica de Catalunya

[3]Aarhus University

[4]McGill University

CSR 2011 — June 14, 2011

This talk

- Boolean functions represented by polynomials over $\mathbf{Z}_m$, with m composite.
- Angluin's exact learning model.
- Programs over groups/monoids.

Modular polynomials

- $P(x_1, \ldots, x_n)$ is a polynomial over $\mathbf{Z}_m$.
- $A \subseteq \mathbf{Z}_m$ is an *accepting set*.

The pair (P, A) computes the Boolean function $f : \{0,1\}^n \to \{0,1\}$ given by

$$f(x) = 1 \quad \text{if and only if} \quad P(x) \in A$$

Modular polynomials

- $P(x_1, \ldots, x_n)$ is a polynomial over $\mathbf{Z}_m$.
- $A \subseteq \mathbf{Z}_m$ is an *accepting set*.

The pair (P, A) computes the Boolean function $f : \{0,1\}^n \to \{0,1\}$ given by

$$f(x) = 1 \quad \text{if and only if} \quad P(x) \in A$$

We also say that P is a "generalized" representation of f.

Modular polynomials

- $P(x_1, \ldots, x_n)$ is a polynomial over $\mathbf{Z}_m$.
- $A \subseteq \mathbf{Z}_m$ is an *accepting set*.

The pair (P, A) computes the Boolean function $f : \{0,1\}^n \to \{0,1\}$ given by

$$f(x) = 1 \quad \text{if and only if} \quad P(x) \in A$$

We also say that P is a "generalized" representation of f.

Most important parameter: The degree $\deg(P)$ of P.

State of affairs

We don’t really understand the computational power of modular polynomials.

State of affairs

We don’t really understand the computational power of modular polynomials.

- “Surprising” upper bounds of degree $O(n^{1/r})$ for some fundamental functions.

State of affairs

We don’t really understand the computational power of modular polynomials.

- “Surprising” upper bounds of degree $O(n^{1/r})$ for some fundamental functions.
- Best degree lower bound is $\Omega(\log n)$.

State of affairs

We don't really understand the computational power of modular polynomials.

- "Surprising" upper bounds of degree $O(n^{1/r})$ for some fundamental functions.
- Best degree lower bound is $\Omega(\log n)$.

This talk: Polynomials of *constant* degree.

Angluin's exact learning model

Let $\mathscr{C}$ be a *concept class* of Boolean functions, together with a scheme for representing the functions.

- Teacher holds Boolean function $f \in \mathscr{C}$ (the *target* function).
- Learner, having no knowledge of f, wishes to *learn* f (i.e find a representation of f).

Angluin's exact learning model

Let $\mathscr{C}$ be a *concept class* of Boolean functions, together with a scheme for representing the functions.

- Teacher holds Boolean function $f \in \mathscr{C}$ (the *target* function).
- Learner, having no knowledge of f, wishes to *learn* f (i.e find a representation of f).

Two types of queries:

Angluin's exact learning model

Let $\mathscr{C}$ be a *concept class* of Boolean functions, together with a scheme for representing the functions.

- Teacher holds Boolean function $f \in \mathscr{C}$ (the *target* function).
- Learner, having no knowledge of f, wishes to *learn* f (i.e find a representation of f).

Two types of queries:

Membership: Learner presents $x \in \{0,1\}^n$. Teacher responds with $f(x)$.

Angluin's exact learning model

Let $\mathscr{C}$ be a *concept class* of Boolean functions, together with a scheme for representing the functions.

- Teacher holds Boolean function $f \in \mathscr{C}$ (the *target* function).
- Learner, having no knowledge of f, wishes to *learn* f (i.e find a representation of f).

Two types of queries:

Membership: Learner presents $x \in \{0,1\}^n$. Teacher responds with $f(x)$.

Equivalence: Learner presents (representation of) Boolean function g. Teacher responds with EQUAL if $f = g$, or presents $x \in \{0,1\}^n$ such that $g(x) \neq f(x)$.

A slight generalization of modular polynomials

- $P(x_1,\ldots,x_n)$ is a polynomial over finite commutative ring with unity $\mathscr{R}$.
- $A \subseteq \mathscr{R}$ is an *accepting set*.

The pair (P,A) computes the Boolean function $f:\{0,1\}^n \to \{0,1\}$ given by

$$f(x) = 1 \quad \text{if and only if} \quad P(x) \in A$$

A slight generalization of modular polynomials

- $P(x_1, \ldots, x_n)$ is a polynomial over finite commutative ring with unity $\mathscr{R}$.
- $A \subseteq \mathscr{R}$ is an *accepting set*.

The pair (P, A) computes the Boolean function $f : \{0,1\}^n \to \{0,1\}$ given by

$$f(x) = 1 \quad \text{if and only if} \quad P(x) \in A$$

Remarks:

- Without loss of generality: $\mathscr{R} = \mathbf{Z}_m^l$ for some m and l.

A slight generalization of modular polynomials

- $P(x_1, \ldots, x_n)$ is a polynomial over finite commutative ring with unity $\mathscr{R}$.
- $A \subseteq \mathscr{R}$ is an *accepting set*.

The pair (P, A) computes the Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ given by

$$f(x) = 1 \quad \text{if and only if} \quad P(x) \in A$$

Remarks:

- Without loss of generality: $\mathscr{R} = \mathbf{Z}_m^l$ for some m and l.
- Equivalently: A constant sized Boolean combination of Boolean functions computed by polynomials over $\mathbf{Z}_m$.

Our goal with this research

Theorem (?)

Let $\mathscr{R}$ be fixed. The class of Boolean functions computed by polynomials over $\mathscr{R}$ of constant degree can be exactly learned in deterministic polynomial from membership queries.

Our goal with this research

Theorem (?)

Let $\mathscr{R}$ be fixed. The class of Boolean functions computed by polynomials over $\mathscr{R}$ of constant degree can be exactly learned in deterministic polynomial from membership queries.

Still an open problem, even when $\mathscr{R} = \mathbf{Z}_m$ for composite m.

Our result

Theorem

Let $\mathscr{R}$ be fixed. The class of Boolean functions computed by read-constant polynomials over $\mathscr{R}$ of constant degree can be exactly learned in deterministic polynomial from membership queries.

Our result

Theorem

Let $\mathscr{R}$ be fixed. The class of Boolean functions computed by read-constant polynomials over $\mathscr{R}$ of constant degree can be exactly learned in deterministic polynomial from membership queries.

Read-constant: Every variable occurs in only constantly many monomials.

Programs over groups/monoids

Let M be a finite monoid.

Instruction Triple $< i, g, h >$, $i \in [n]$, $g, h \in M$.

Program List $L = (\ell_1, \ldots, \ell_m)$ of instructions, and *accepting set* $A \subseteq M$.

Programs over groups/monoids

Let M be a finite monoid.

Instruction Triple $< i, g, h >$, $i \in [n]$, $g, h \in M$.

Program List $L = (\ell_1, \ldots, \ell_m)$ of instructions, and *accepting set* $A \subseteq M$.

The pair (L, A) computes the Boolean function function f as follows.

$$< i, g, h >(x) = \begin{cases} g & \text{if } x_i = 1 \\ h & \text{if } x_i = 0 \end{cases}$$

Programs over groups/monoids

Let M be a finite monoid.

Instruction Triple $< i, g, h >$, $i \in [n]$, $g, h \in M$.

Program List $L = (\ell_1, \ldots, \ell_m)$ of instructions, and *accepting set* $A \subseteq M$.

The pair (L, A) computes the Boolean function function f as follows.

$$< i, g, h >(x) = \begin{cases} g & \text{if } x_i = 1 \\ h & \text{if } x_i = 0 \end{cases}$$

$$f(x) = 1 \quad \text{if and only if} \quad \prod_{i=1}^{m} \ell_i(x) \in A$$

Programs vs. circuit classes

Algebraic structure	Circuit class
Any monoid	NC^1
Solvable monoid	ACC^0
Aperiodic monoid	AC^0
⋮	

Programs vs. circuit classes

Algebraic structure	Circuit class
Any monoid	NC^1
Solvable monoid	ACC^0
Aperiodic monoid	AC^0
⋮	
Solvable group	CC^0
Nilpotent group	$NC^0 \circ MOD \circ NC^0$
Abelian group	$NC^0 \circ MOD$
p-group	$MOD_p \circ NC^0$
⋮	

Programs vs. circuit classes

Algebraic structure	Circuit class
Any monoid	NC^1
Solvable monoid	ACC^0
Aperiodic monoid	AC^0
$\vdots$	
Solvable group	CC^0
Nilpotent group	$NC^0 \circ MOD \circ NC^0$
Abelian group	$NC^0 \circ MOD$
p-group	$MOD_p \circ NC^0$
$\vdots$	

The class of Boolean functions computed by constant degree polynomials over a finite commutative ring with unity $\mathscr{R}$ is a very natural concept class!

Notation

- $M \subseteq [n]$ represent the monomial $\prod_{i \in M} x_i$.
- $\chi_M \in \{0,1\}^n$ is characteristic vector of M.
- c_M is coefficient of monomial M.
- If $w \in \{0,1\}^n$, $I_w \subseteq [n]$ is the indices i where $w_i = 1$.

Structural properties of polynomials

Theorem (Péladeau and Thérien)

Let $\mathscr{R}$ be a finite commutative ring with unity and d any number. Then there exist a constant $c = c(\mathscr{R}, d)$ such that:

- *For any polynoimial P over $\mathscr{R}$ of degree at most d and for any $r \in \mathrm{range}(P)$ there exist $w \in \{0,1\}^n$ with $|I_w| \leq c$ such that $P(w) = r$.*

Structural properties of polynomials

Theorem (Péladeau and Thérien)

Let $\mathscr{R}$ be a finite commutative ring with unity and d any number. Then there exist a constant $c = c(\mathscr{R}, d)$ such that:

- *For any polynoimial P over $\mathscr{R}$ of degree at most d and for any $r \in \mathrm{range}(P)$ there exist $w \in \{0,1\}^n$ with $|I_w| \leq c$ such that $P(w) = r$.*

Remarks:

- Proved using an inductive Ramsey-theoretic argument.
- Strong relation to the degree for representing the AND function over $\mathscr{R}$.

Boolean function determined by low-weight inputs

Corollary

There exist a constant $c' = c'(\mathscr{R}, d)$ such that:

- *Let P and Q be polynomials of degree at most d with accepting sets A and B.*
- *If the Boolean functions computed by the pairs (P, A) and (Q, B) agree on all inputs $w \in \{0,1\}^n$ with $|I_w| \leq c'$, then the two Boolean functions are identical.*

Boolean function determined by low-weight inputs

Corollary

There exist a constant $c' = c'(\mathscr{R}, d)$ *such that:*

- *Let* P *and* Q *be polynomials of degree at most* d *with accepting sets* A *and* B*.*
- *If the Boolean functions computed by the pairs* (P, A) *and* (Q, B) *agree on all inputs* $w \in \{0,1\}^n$ *with* $|I_w| \leq c'$*, then the two Boolean functions are identical.*

Proof.

Let $c' = c(\mathscr{R} \times \mathscr{R}, d)$. Consider the polynomial $P \times Q$ with P and Q as separate coordinates. Any value of the range is assumed on input of weight at most c'. □

A consistency test

Input: Polynomial Q with accepting set A. Membership query access to Boolean function f.
Task: Decide if the pair (Q, A) computes the function f.

A consistency test

Input: Polynomial Q with accepting set A. Membership query access to Boolean function f.
Task: Decide if the pair (Q,A) computes the function f.

- Query f on all $w \in \{0,1\}^n$ with $|I_w| \leq c'(\mathscr{R},d)$
- Return true if and only if for each queried w, $f(w)=1$ if and only if $Q(w) \in A$

A consistency test

Input: Polynomial Q with accepting set A. Membership query access to Boolean function f.
Task: Decide if the pair (Q, A) computes the function f.

- Query f on all $w \in \{0,1\}^n$ with $|I_w| \leq c'(\mathscr{R}, d)$
- Return true if and only if for each queried w, $f(w) = 1$ if and only if $Q(w) \in A$

This also gives equivalence queries for free.

A template for learning

- Define a search space: Let $\mathscr{P}$ be guaranteed to include the target ($\mathscr{P}$ presumably depends on answers to membership queries).

A template for learning

- Define a search space: Let $\mathscr{P}$ be guaranteed to include the target ($\mathscr{P}$ presumably depends on answers to membership queries).
- Exhaustive search: for each $P \in \mathscr{P}$ perform consistency test on P, return first P that passes.

Difficulties

- Different polynomials can represent the same Boolean function.

Difficulties

- Different polynomials can represent the same Boolean function.
- A query χ_M with $|M| = 1$ only reveals whether $c_M \in A$.

Difficulties

- Different polynomials can represent the same Boolean function.
- A query χ_M with $|M| = 1$ only reveals whether $c_M \in A$.
- A query χ_M in general only reveals whether $\sum_{I \subseteq M} c_I \in A$.

Difficulties

- Different polynomials can represent the same Boolean function.
- A query χ_M with $|M| = 1$ only reveals whether $c_M \in A$.
- A query χ_M in general only reveals whether $\sum_{I \subseteq M} c_I \in A$.

Our approach: Define an equivalence relation on the monomials, such that equivalent monomials can be assumed to have same coefficient.

An attempt

Let M and M' be of degree d.
Say $M \equiv_d M'$ if:

- $\forall r \in \mathscr{R} : r + P(\chi_M) \in A$ if and only if $r + P(\chi_{M'}) \in A$.
- For all immediate sub-monomials $M_1, \ldots, M_d$ and $M'_1, \ldots, M'_d$ we have $M_i \equiv_{d-1} M'_i$.

Suppose we know the equivalence classes of $\equiv_d$'s and let $\mathscr{P}$ be the set of polynomials respecting these.

An attempt

Let M and M' be of degree d.
Say $M \equiv_d M'$ if:

- $\forall r \in \mathscr{R} : r + P(\chi_M) \in A$ if and only if $r + P(\chi_{M'}) \in A$.
- For all immediate sub-monomials $M_1, \ldots, M_d$ and $M'_1, \ldots, M'_d$ we have $M_i \equiv_{d-1} M'_i$.

Suppose we know the equivalence classes of $\equiv_d$'s and let $\mathscr{P}$ be the set of polynomials respecting these.

- Good: Small search space (in fact constant size).

An attempt

Let M and M' be of degree d.
Say $M \equiv_d M'$ if:

- $\forall r \in \mathscr{R} : r + P(\chi_M) \in A$ if and only if $r + P(\chi_{M'}) \in A$.
- For all immediate sub-monomials $M_1, \ldots, M_d$ and $M'_1, \ldots, M'_d$ we have $M_i \equiv_{d-1} M'_i$.

Suppose we know the equivalence classes of $\equiv_d$'s and let $\mathscr{P}$ be the set of polynomials respecting these.

- Good: Small search space (in fact constant size).
- Good: Definition looks a little like membership queries...

An attempt

Let M and M' be of degree d.
Say $M \equiv_d M'$ if:

- $\forall r \in \mathscr{R} : r + P(\chi_M) \in A$ if and only if $r + P(\chi_{M'}) \in A$.
- For all immediate sub-monomials $M_1, \ldots, M_d$ and $M'_1, \ldots, M'_d$ we have $M_i \equiv_{d-1} M'_i$.

Suppose we know the equivalence classes of $\equiv_d$'s and let $\mathscr{P}$ be the set of polynomials respecting these.

- Good: Small search space (in fact constant size).
- Good: Definition looks a little like membership queries...
- Bad: but not exactly, so assumption is probably unrealistic!

An attempt

Let M and M' be of degree d.
Say $M \equiv_d M'$ if:

- $\forall r \in \mathscr{R} : r + P(\chi_M) \in A$ if and only if $r + P(\chi_{M'}) \in A$.
- For all immediate sub-monomials $M_1, \ldots, M_d$ and $M'_1, \ldots, M'_d$ we have $M_i \equiv_{d-1} M'_i$.

Suppose we know the equivalence classes of $\equiv_d$'s and let $\mathscr{P}$ be the set of polynomials respecting these.

- Good: Small search space (in fact constant size).
- Good: Definition looks a little like membership queries...
- Bad: but not exactly, so assumption is probably unrealistic!
- Importantly: Does it include the target function?

Changing coefficients

Suppose $M \equiv_d M'$:

- $\forall r \in \mathscr{R} : r + P(\chi_M) \in A$ if and only if $r + P(\chi_{M'}) \in A$.
- For all immediate sub-monomials $M_1, \ldots, M_d$ and $M'_1, \ldots, M'_d$ we have $M_i \equiv_{d-1} M'_i$.

Changing coefficients

Suppose $M \equiv_d M'$:

- $\forall r \in \mathscr{R} : r + P(\chi_M) \in A$ if and only if $r + P(\chi_{M'}) \in A$.
- For all immediate sub-monomials $M_1, \ldots, M_d$ and $M'_1, \ldots, M'_d$ we have $M_i \equiv_{d-1} M'_i$.

Suppose (by induction?) that $c_{M_i} = c_{M'_i}$ for all i, and define $P' = P[c_{M'} \leftarrow c_M]$.

Changing coefficients

Suppose $M \equiv_d M'$:

- $\forall r \in \mathscr{R} : r + P(\chi_M) \in A$ if and only if $r + P(\chi_{M'}) \in A$.
- For all immediate sub-monomials $M_1, \ldots, M_d$ and $M'_1, \ldots, M'_d$ we have $M_i \equiv_{d-1} M'_i$.

Suppose (by induction?) that $c_{M_i} = c_{M'_i}$ for all i, and define $P' = P[c_{M'} \leftarrow c_M]$.

- Let $x \in \{0,1\}^n$ be arbitrary.
- We need only consider x such that $M' \subseteq I_x$.

Changing coefficients

Suppose $M \equiv_d M'$:

- $\forall r \in \mathscr{R} : r + P(\chi_M) \in A$ if and only if $r + P(\chi_{M'}) \in A$.
- For all immediate sub-monomials $M_1, \ldots, M_d$ and $M'_1, \ldots, M'_d$ we have $M_i \equiv_{d-1} M'_i$.

Suppose (by induction?) that $c_{M_i} = c_{M'_i}$ for all i, and define $P' = P[c_{M'} \leftarrow c_M]$.

- Let $x \in \{0,1\}^n$ be arbitrary.
- We need only consider x such that $M' \subseteq I_x$.
- Define $r = P(x) - P(\chi_{M'})$.

Changing coefficients

Suppose $M \equiv_d M'$:

- $\forall r \in \mathscr{R} : r + P(\chi_M) \in A$ if and only if $r + P(\chi_{M'}) \in A$.
- For all immediate sub-monomials $M_1, \ldots, M_d$ and $M'_1, \ldots, M'_d$ we have $M_i \equiv_{d-1} M'_i$.

Suppose (by induction?) that $c_{M_i} = c_{M'_i}$ for all i, and define $P' = P[c_{M'} \leftarrow c_M]$.

- Let $x \in \{0,1\}^n$ be arbitrary.
- We need only consider x such that $M' \subseteq I_x$.
- Define $r = P(x) - P(\chi_{M'})$.
- $r + P(\chi_{M'}) = P(x) - P(\chi_{M'}) + P(\chi_{M'}) = P(x)$.

Changing coefficients

Suppose $M \equiv_d M'$:

- $\forall r \in \mathscr{R} : r + P(\chi_M) \in A$ if and only if $r + P(\chi_{M'}) \in A$.
- For all immediate sub-monomials $M_1, \ldots, M_d$ and $M'_1, \ldots, M'_d$ we have $M_i \equiv_{d-1} M'_i$.

Suppose (by induction?) that $c_{M_i} = c_{M'_i}$ for all i, and define $P' = P[c_{M'} \leftarrow c_M]$.

- Let $x \in \{0,1\}^n$ be arbitrary.
- We need only consider x such that $M' \subseteq I_x$.
- Define $r = P(x) - P(\chi_{M'})$.
- $r + P(\chi_{M'}) = P(x) - P(\chi_{M'}) + P(\chi_{M'}) = P(x)$.
- $r + P(\chi_M) = P(x) - P(\chi_{M'}) + P(\chi_M) = P(x) - P(\chi_{M'}) + P'(\chi_{M'}) = P'(x)$.

Changing coefficients

Suppose $M \equiv_d M'$:

- $\forall r \in \mathscr{R} : r + P(\chi_M) \in A$ if and only if $r + P(\chi_{M'}) \in A$.
- For all immediate sub-monomials $M_1, \ldots, M_d$ and $M'_1, \ldots, M'_d$ we have $M_i \equiv_{d-1} M'_i$.

Suppose (by induction?) that $c_{M_i} = c_{M'_i}$ for all i, and define $P' = P[c_{M'} \leftarrow c_M]$.

- Let $x \in \{0,1\}^n$ be arbitrary.
- We need only consider x such that $M' \subseteq I_x$.
- Define $r = P(x) - P(\chi_{M'})$.
- $r + P(\chi_{M'}) = P(x) - P(\chi_{M'}) + P(\chi_{M'}) = P(x)$.
- $r + P(\chi_M) = P(x) - P(\chi_{M'}) + P(\chi_M) =$
  $P(x) - P(\chi_{M'}) + P'(\chi_{M'}) = P'(x)$.

Conclusion: P and P' compute the same function.

A problem

- The equivalence relation defined is highly representation dependent.

A problem

- The equivalence relation defined is highly representation dependent.
- Changing coefficients may possibly change the equivalence classes as well!

A problem

- The equivalence relation defined is highly representation dependent.
- Changing coefficients may possibly change the equivalence classes as well!
- Our real equivalence relation is defined only in terms of the function represented together with a property of the polynomial that is preserved by altering coefficients (in the similar way)!

Recall: Structural properties of polynomials

Theorem (Péladeau and Thérien)

Let $\mathscr{R}$ be a finite commutative ring with unity and d any number. Then there exist a constant $c = c(\mathscr{R}, d)$ such that:

- *For any polynoimial P over $\mathscr{R}$ of degree at most d and for any $r \in \text{range}(P)$ there exist $w \in \{0,1\}^n$ with $|I_w| \leq c$ such that $P(w) = r$.*

Magic sets for polynomials

Corollary

There exist a constant $s = s(\mathscr{R}, d)$, such that for every multilinear polynomial P over $\mathscr{R}$ of degree at most d, there exist a set $J \subset \{1, \ldots, n\}$ with the following properties:

- $|J| \leq s$.
- *For every $r \in \text{range}(P)$ there exist $w \in \{0,1\}^n$ with $I_w \subseteq J$ such that $P(w) = r$.*

We call J a “magic set” for P.

Using a magic set

- Assume P has magic set J.

Using a magic set

- Assume P has magic set J.
- Let K be variables sharing monomial with variable in J.
- Let $N = [n] \setminus (J \cup K)$. (We say N is at distance ≥ 2 from J).

Using a magic set

- Assume P has magic set J.
- Let K be variables sharing monomial with variable in J.
- Let $N = [n] \setminus (J \cup K)$. (We say N is at distance ≥ 2 from J).
- Let $w, x \in \{0,1\}^n$ with $I_w \subseteq J, I_x \subseteq N$.

Using a magic set

- Assume P has magic set J.
- Let K be variables sharing monomial with variable in J.
- Let $N = [n] \setminus (J \cup K)$. (We say N is at distance ≥ 2 from J).
- Let $w, x \in \{0,1\}^n$ with $I_w \subseteq J, I_x \subseteq N$.
- Then $P(w \vee x) = P(w) + P(x)$.

Using a magic set

- Assume P has magic set J.
- Let K be variables sharing monomial with variable in J.
- Let $N = [n] \setminus (J \cup K)$. (We say N is at distance ≥ 2 from J).
- Let $w, x \in \{0,1\}^n$ with $I_w \subseteq J, I_x \subseteq N$.
- Then $P(w \vee x) = P(w) + P(x)$.

Note: When P is *read-constant* also K is of constant size.

Properties of polynomials with magic sets

Lemma

Let

- $P(x) = \sum_{I \subseteq [n], |I| \leq d} c_I \prod_{i \in I} x_i$ *be any polynomial over* $\mathscr{R}$, *with a magic set* J.
- *Let* N *be the set of indices at distance* ≥ 2 *from* J.

Then

- $r + \sum_{I \subseteq N, |I| \leq d} \lambda_I c_I \in \text{range}(P)$, *for all* $r \in \text{range}(P)$ *and all* $\lambda_I \in \{0, \ldots, |\mathscr{R}| - 1\}$.

Properties of polynomials with magic sets

Lemma

Let

- $P(x) = \sum_{I \subseteq [n], |I| \leq d} c_I \prod_{i \in I} x_i$ *be any polynomial over* $\mathscr{R}$, *with a magic set* J.
- *Let* N *be the set of indices at distance* ≥ 2 *from* J.

Then

- $r + \sum_{I \subseteq N, |I| \leq d} \lambda_I c_I \in \text{range}(P)$, *for all* $r \in \text{range}(P)$ *and all* $\lambda_I \in \{0, \ldots, |\mathscr{R}| - 1\}$.

Proof.

By induction, repeatedly moving supports of inputs to J. □

The real attempt

Assume J is magic set for P, and let M and M' be of degree d.
Say $M \equiv_{d,J} M'$ if:

- $\forall w, I_w \subseteq J : P(w \vee \chi_M) \in A$ if and only if $P(w \vee \chi_{M'}) \in A$.
- For all immediate sub-monomials $M_1, \ldots, M_d$ and $M'_1, \ldots, M'_d$ we have $M_i \equiv_{d-1,J} M'_i$.

The real attempt

Assume J is magic set for P, and let M and M' be of degree d. Say $M \equiv_{d,J} M'$ if:

- $\forall w, I_w \subseteq J : P(w \vee \chi_M) \in A$ if and only if $P(w \vee \chi_{M'}) \in A$.
- For all immediate sub-monomials $M_1, \ldots, M_d$ and $M'_1, \ldots, M'_d$ we have $M_i \equiv_{d-1,J} M'_i$.

If $I_M, I'_M \subseteq N$, we can change coefficients as before, and the induction now works.

The search space

Let $\mathscr{P}$ be the set of polynomials satisfying:
There is a magic set J, such that for all monomials M, M' of degree d with $I_M, I'_M \subseteq N$, if $M \equiv_{d,J} M'$ then $c_M = c'_M$.

The search space

Let $\mathscr{P}$ be the set of polynomials satisfying:
There is a magic set J, such that for all monomials M, M' of degree d with $I_M, I'_M \subseteq N$, if $M \equiv_{d,J} M'$ then $c_M = c'_M$.

Note: Given J and K, equivalence classes of $\equiv_{d,J}$ can be computed using membership queries. Remains to specify:

- Coefficients of monomials containing variables from $J \cup K$.
- Coefficients of equivalence classes.

The search space

Let $\mathscr{P}$ be the set of polynomials satisfying:
There is a magic set J, such that for all monomials M, M' of degree d with $I_M, I'_M \subseteq N$, if $M \equiv_{d,J} M'$ then $c_M = c'_M$.

Note: Given J and K, equivalence classes of $\equiv_{d,J}$ can be computed using membership queries. Remains to specify:

- Coefficients of monomials containing variables from $J \cup K$.
- Coefficients of equivalence classes.

Conclusion:

- $\mathscr{P}$ is of polynomial size, and can be exhaustively searched using membership queries!

Extensions to higher degree

Using a result of Tardos and Barrington on the MOD_m degree of AND, we can show

$$c(\mathbf{Z}_m^l, d), c'(\mathbf{Z}_m^l, d), s(\mathbf{Z}_m^l, d) \leq \gamma^{d^{r-1}}$$

where r is the number of distinct prime divisors of m, and γ depends on m and l.

Extensions to higher degree

Using a result of Tardos and Barrington on the MOD_m degree of AND, we can show

$$c(\mathbf{Z}_m^l, d), c'(\mathbf{Z}_m^l, d), s(\mathbf{Z}_m^l, d) \leq \gamma^{d^{r-1}}$$

where r is the number of distinct prime divisors of m, and γ depends on m and l.

Result: Sub-exponential learning algorithm for slowly growing d and k for learning degree d read-k polynomials.

Conclusion

New Result:

- A Polynomial time learning algorithm for read-constant polynomials of constant degree over finite rings.

Conclusion

New Result:

- A Polynomial time learning algorithm for read-constant polynomials of constant degree over finite rings.

Questions:

- Can we avoid the read-constant assumption?
- Other uses of “magic sets”?