Polynomial Bounds for Invariant Functions Separating Orbits

Harlan Kadish

University of Michigan

July 3, 2010

Outline

- Briefing on Separating Orbits
- A New Algorithm
- Complexity via Straight Line Programs
- How the Algorithm Works

Briefing on Separating Orbits

Let G be an algebraic group acting rationally on a variety V.

Definition

The **orbit** *of a point* $x \in V$ *is the set*

$$G \cdot x = \{g \cdot x \mid g \in G\}.$$

1. If $x, y \in V$, can we find out if x and y lie in the same orbit?
2. How easily can we find out?

Question (1) is asked and answered:

- Applications include structural chemistry, computer vision, and dynamical systems.
- Potentially answered by the invariant subring,

$$k[V]^G = \{f(p) \in k[V] \mid f(g^{-1} \cdot p) = f(p) \ \forall\, g \in G\}$$

Briefing on Separating Orbits

Let G be an algebraic group acting rationally on a variety V.

Definition

The **orbit** *of a point $x \in V$ is the set*

$$G \cdot x = \{g \cdot x \mid g \in G\}.$$

1. If $x, y \in V$, can we find out if x and y lie in the same orbit?
2. How easily can we find out?

Question (1) is asked and answered:

- Applications include structural chemistry, computer vision, and dynamical systems.
- Potentially answered by the invariant subring,

$$k[V]^G = \{f(p) \in k[V] \mid f(g^{-1} \cdot p) = f(p) \; \forall \, g \in G\}$$

Briefing on Separating Orbits

Definition

A set S of invariant functions on V **separates orbits** *if whenever $x \notin G \cdot y$, then $\exists\, f \in S$ such that $f(x) \neq f(y)$.*

- If G is reductive,
  - $k[V]^G$ is finitely generated, so generators may separate orbits.
  - Can compute generators using Gröbner bases.
- If G not reductive, still $\exists$ finite $S \subset k[V]^G$ such that for each $x, y \in V$,

  If $\exists\, h \in k[V]^G$ such that $h(x) \neq h(y)$,
  Then $\exists\, f \in S$ such that $f(x) \neq f(y)$.
- So S separates orbits as precisely as $k[V]^G$.

Briefing on Separating Orbits

Definition

A set S of invariant functions on V **separates orbits** *if whenever $x \notin G \cdot y$, then $\exists\, f \in S$ such that $f(x) \neq f(y)$.*

- If G is reductive,
  - $k[V]^G$ is finitely generated, so generators may separate orbits.
  - Can compute generators using Gröbner bases.
- If G not reductive, still $\exists$ finite $S \subset k[V]^G$ such that for each $x, y \in V$,

  If $\exists\, h \in k[V]^G$ such that $h(x) \neq h(y)$,
  Then $\exists\, f \in S$ such that $f(x) \neq f(y)$.
- So S separates orbits as precisely as $k[V]^G$.

Briefing on Separating Orbits

Definition

A set S of invariant functions on V **separates orbits** *if whenever $x \notin G \cdot y$, then $\exists\, f \in S$ such that $f(x) \neq f(y)$.*

- If G is reductive,
  - $k[V]^G$ is finitely generated, so generators may separate orbits.
  - Can compute generators using Gröbner bases.
- If G not reductive, still $\exists$ finite $S \subset k[V]^G$ such that for each $x, y \in V$,

  If $\exists\, h \in k[V]^G$ such that $h(x) \neq h(y)$,
  Then $\exists\, f \in S$ such that $f(x) \neq f(y)$.
- So S separates orbits as precisely as $k[V]^G$.

Limitations of $k[V]^G$, Part 1

Limitations of theory: regular functions may fail to separate orbits.

- Let $\mathbb{G}_m = k^*$ act on $\mathbb{A}^2$ by

$$g \cdot (x, y) = (gx, gy).$$

- Then $k[x, y]^{\mathbb{G}_m} = k$.
- In general, failure when $\exists\, z \in \overline{G \cdot x} \cap \overline{G \cdot y} \neq \emptyset$:
- For if $f \in k[V]^G$, then $f(G \cdot x) = f(z) = f(G \cdot y)$.

Limitations of $k[V]^G$, Part 2

Limitations of practice:

- Gröbner basis calculations are costly in principle.
- Only have algorithms for S or $k[V]^G$ generators if G reductive.
- For general G, can't predict number of separating or generating invariants.

A New Breed of Function

Extend the regular functions on V with a **quasi-inverse**:

$$\{f\}(p) = \begin{cases} 1/f(p) & f(p) \neq 0 \\ 0 & f(p) = 0 \end{cases}$$

Definition

For $R = k[V]$, let $\widehat{R}$ denote the ring of functions $V \to k$ obtained by applying the quasi-inverse iteratively on elements of R. Call these functions **constructible***.*

E.g., if $f, g \in R$, then $\{f + \{g\}\} \in \widehat{R}$.

A New Algorithm for Separating Orbits

- Over $k = \overline{k}$, let $G \hookrightarrow \mathbb{A}^\ell$ be an m-dimensional algebraic group.
- Let G act rationally on $\mathbb{A}^n$ via the representation $\rho \colon G \hookrightarrow GL_n$.
- Let $N = \max\{\deg(\rho_{ij})\}$.
- Let r be the maximal dimension of an orbit.

Theorem

There is an algorithm to produce a finite set $\mathcal{C} \subset \widehat{R}$ of invariant, constructible functions with the following properties:

1. *The set $\mathcal{C}$ separates orbits.*
2. *The size of $\mathcal{C}$ grows as $O(n^2 N^{(\ell+m+1)(r+1)})$.*
3. *The $f \in \mathcal{C}$ can be written as straight line programs, such that the sum of their lengths is $O(n^3 N^{3\ell(r+1)+r})$.*

A New Algorithm for Separating Orbits

- Over $k = \overline{k}$, let $G \hookrightarrow \mathbb{A}^{\ell}$ be an m-dimensional algebraic group.
- Let G act rationally on $\mathbb{A}^n$ via the representation $\rho\colon G \hookrightarrow GL_n$.
- Let $N = \max\{\deg(\rho_{ij})\}$.
- Let r be the maximal dimension of an orbit.

Theorem

There is an algorithm to produce a finite set $\mathcal{C} \subset \widehat{R}$ of invariant, constructible functions with the following properties:

1. *The set $\mathcal{C}$ separates orbits.*
2. *The size of $\mathcal{C}$ grows as $O(n^2 N^{(\ell+m+1)(r+1)})$.*
3. *The $f \in \mathcal{C}$ can be written as straight line programs, such that the sum of their lengths is $O(n^3 N^{3\ell(r+1)+r})$.*

Example

Let $\mathbb{G}_m = k^*$ act on $\mathbb{A}^2$ by

$$g \cdot (x, y) = (gx, gy), \quad \text{so} \quad k[x, y]^{\mathbb{G}_m} = k.$$

The functions in $\mathcal{C}$ simplify to

$$x\{x\} \quad \text{and} \quad y\{y\} \cdot (1 - x\{x\} + y\{x\}).$$

$$\text{Recall } \{f\}(p) = \begin{cases} 1/f(p) & f(p) \neq 0 \\ 0 & f(p) = 0 \end{cases}$$

- If $x \neq 0$, then $x\{x\} = x/x = 1$ and $y\{x\} = y/x$.
- Invariance: $x \neq 0 \implies (gx)\{gx\} = 1$, $gy\{gx\} = y/x$,
- Separation:

$$x, y \neq 0 \implies y\{y\} \cdot (1 - x\{x\} + y\{x\}) = 1 \cdot (1 - 1 + y/x) = y/x.$$

What Will It Cost Me?

- The theorem says more than the existence of $\mathcal{C}$:

$$|\mathcal{C}| = O\left(n^2 N^{(\ell+m+1)(r+1)}\right)$$

- Still, how practical is it to use $\mathcal{C}$?
  - How long does it take to write down the functions?
  - How complicated is the evaluation of the functions?

What Will It Cost Me?

- The theorem says more than the existence of $\mathcal{C}$:

$$|\mathcal{C}| = O\left(n^2 N^{(\ell+m+1)(r+1)}\right)$$

- Still, how practical is it to use $\mathcal{C}$?
    - How long does it take to write down the functions?
    - How complicated is the evaluation of the functions?

Straight Line Programs

Definition

An **SLP** *is a finite list of ring operations (and the quasi-inverse) to perform on a finite input sequence of ring elements.*

- E.g., write $x\{y\} + \{z\}$ as an SLP:
  1. Input (x, y, z).
  2. Compute $\{y\}$.
  3. Multiply x and $\{y\}$.
  4. Compute $\{z\}$.
  5. Add $x\{y\}$ to $\{z\}$.
- Output is a sequence: $(x, y, z, \{y\}, x\{y\}, \{z\}, x\{y\} + \{z\})$.

Definition

The **complexity** *of an SLP is the non-input length of its output.*

Straight Line Programs

Definition

An **SLP** *is a finite list of ring operations (and the quasi-inverse) to perform on a finite input sequence of ring elements.*

- E.g., write $x\{y\} + \{z\}$ as an SLP:
  1. Input (x, y, z).
  2. Compute $\{y\}$.
  3. Multiply x and $\{y\}$.
  4. Compute $\{z\}$.
  5. Add $x\{y\}$ to $\{z\}$.
- Output is a sequence: $(x, y, z, \{y\}, x\{y\}, \{z\}, x\{y\} + \{z\})$.

Definition

The **complexity** *of an SLP is the non-input length of its output.*

Separating Orbits Cheaply

Theorem

There is an algorithm to produce a finite set $\mathcal{C} \subset \widehat{R}$ of invariant, constructible functions with the following properties:

1. *The set $\mathcal{C}$ separates orbits.*
2. *The size of $\mathcal{C}$ grows as $O(n^2 N^{(\ell+m+1)(r+1)})$.*
3. *The $f \in \mathcal{C}$ can be written as straight line programs, such that the sum of their lengths is $O(n^3 N^{3\ell(r+1)+r})$.*

- Can write down $\mathcal{C}$ for *any algebraic group.*
- Have a polynomial bound on $|\mathcal{C}|$.
- Number of steps to write down $\mathcal{C}$ has a polynomial bound.
- Or, can evaluate all of $\mathcal{C}$ at $p \in \mathbb{A}^n$ in polynomial time.

The Algorithm: The Ideal of $G \cdot p$

Fix $p \in \mathbb{A}^n$. To compute defining equations for the closure $\overline{G \cdot p}$,

1. From $\rho : G \to GL_n$, write down the orbit map

$$\sigma_p \colon G \to \mathbb{A}^n \quad \text{defined by} \quad \sigma_p \colon g \mapsto \rho(g) \cdot p.$$

2. Write down the ring map $\sigma_p^* \colon k[x_1, \ldots, x_n] \to k[G]$.
3. Then $\ker \sigma_p^*$ is the ideal vanishing on $G \cdot p$.

The Algorithm: Computing $\ker \sigma_p^*$

Lemma

For fixed G, there exists an integer $d = d(N)$, polynomial in N, such that $\overline{G \cdot p}$ can be defined by polynomials of degree $\leq d$.

1. Let $(\sigma_p^*)_{\leq d}$ denote a matrix for the k-vector space map

$$k[x_1, \ldots, x_n]_{\leq d} \to k[G],$$

$$k[x]_{\leq d} = \{f \in k[x] \mid \deg(f) \leq d\}$$

   where the basis on the left is $x_1, \ldots, x_n, x_1^2, x_1 x_2, \ldots, x_n^d$.
2. Basis vectors in the kernel give relations on the monomials of $k[x_1, \ldots, x_n]$.
3. These polynomials would define $\overline{G \cdot p}$.

The Algorithm: Computing $\ker \sigma_p^*$

Lemma

For fixed G, there exists an integer $d = d(N)$, polynomial in N, such that $\overline{G \cdot p}$ can be defined by polynomials of degree $\leq d$.

1. Let $(\sigma_p^*)_{\leq d}$ denote a matrix for the k-vector space map

$$k[x_1, \ldots, x_n]_{\leq d} \to k[G],$$

$$k[x]_{\leq d} = \{f \in k[x] \mid \deg(f) \leq d\}$$

   where the basis on the left is $x_1, \ldots, x_n, x_1^2, x_1 x_2, \ldots, x_n^d$.
2. Basis vectors in the kernel give relations on the monomials of $k[x_1, \ldots, x_n]$.
3. These polynomials would define $\overline{G \cdot p}$.

The Algorithm: Controlling Monomials

A problem arises:

- The dimension of the k-basis

$$x_1, \ldots, x_n,\ x_1^2,\ x_1x_2,\ x_1x_3, \ldots, x_n^d$$

  grows exponentially in n.

- Instead, for every degree $i = 1, \ldots, d$,
  1. Compute the reduced row echelon form of $(\sigma_p^*)_{\leq i}$.
  2. Compute the kernel of $(\sigma_p^*)_{\leq i}$.
  3. Find a maximal set of monomials $M_i \subset k[x_1, \ldots x_n]_{\leq i}$ with linearly independent images in $k[G]$.
  4. Write $(\sigma_p^*)_{\leq (i+1)}$ in terms of M_i and

$$\{m \cdot x_j \mid m \in M_i,\ j = 1 \ldots, n\}.$$

- From Hilbert polynomial of G, know $|M_i|$ is polynomial in i.

The Algorithm: Controlling Monomials

A problem arises:

- The dimension of the k-basis

$$x_1, \ldots, x_n,\ x_1^2,\ x_1x_2,\ x_1x_3, \ldots, x_n^d$$

  grows exponentially in n.
- Instead, for every degree $i = 1, \ldots, d$,
  1. Compute the reduced row echelon form of $(\sigma_p^*)_{\leq i}$.
  2. Compute the kernel of $(\sigma_p^*)_{\leq i}$.
  3. Find a maximal set of monomials $M_i \subset k[x_1, \ldots x_n]_{\leq i}$ with linearly independent images in $k[G]$.
  4. Write $(\sigma_p^*)_{\leq (i+1)}$ in terms of M_i and

$$\{m \cdot x_j \mid m \in M_i,\ j = 1 \ldots, n\}.$$

- From Hilbert polynomial of G, know $|M_i|$ is polynomial in i.

The Algorithm: Enter Constructible Functions

1. Now, the degree bound d determines the dimensions of the matrices $(\sigma_p^*)_{\leq i}$.
2. For fixed G, the degree bound $d = d(N)$ is polynomial in $N = \max\{\deg(\rho_{ij})\}$.
3. Hence the dimensions of the $(\sigma_p^*)_{\leq i}$ have polynomial bounds in n and N.

Proposition

If A is an $s \times t$ matrix, then there exists an SLP (involving the quasi-inverse) for the reduced row echelon form and kernel of A, with complexity $O(st^2 + t^3)$.

4. So we can compute the $\ker(\sigma_p^*)_{\leq i}$ in polynomial time.

The Algorithm: Enter Constructible Functions

1. Now, the degree bound d determines the dimensions of the matrices $(\sigma_p^*)_{\leq i}$.
2. For fixed G, the degree bound $d = d(N)$ is polynomial in $N = \max\{\deg(\rho_{ij})\}$.
3. Hence the dimensions of the $(\sigma_p^*)_{\leq i}$ have polynomial bounds in n and N.

Proposition

If A is an $s \times t$ matrix, then there exists an SLP (involving the quasi-inverse) for the reduced row echelon form and kernel of A, with complexity $O(st^2 + t^3)$.

4. So we can compute the $\ker(\sigma_p^*)_{\leq i}$ in polynomial time.

The Algorithm: Output!

1. For $p \in \mathbb{A}^n$, write down the orbit map $\sigma_p : G \to \mathbb{A}^n$.
2. Write down matrices for $\sigma_p^*\colon k[x_1, \ldots, x_n]_{\leq i} \to k[G]$ up to degree d.
3. Now, the matrix entries are regular functions of p.
4. So the entries of the $\ker(\sigma_p^*)_{\leq i}$ vectors are constructible functions of p.
5. Collect the kernel vectors' entries into the set $\mathcal{C}$.
6. As functions of p, they are G-invariant and separate orbits.
7. Their number and complexity are polynomial in n and N.

The Algorithm: Output!

1. For $p \in \mathbb{A}^n$, write down the orbit map $\sigma_p : G \to \mathbb{A}^n$.
2. Write down matrices for $\sigma_p^*\colon k[x_1, \ldots, x_n]_{\leq i} \to k[G]$ up to degree d.
3. Now, the matrix entries are regular functions of p.
4. So the entries of the $\ker(\sigma_p^*)_{\leq i}$ vectors are constructible functions of p.
5. Collect the kernel vectors' entries into the set $\mathcal{C}$.
6. As functions of p, they are G-invariant and separate orbits.
7. Their number and complexity are polynomial in n and N.