Invariants of Matrix Tuples

Madeline Brandt

March 14, 2019

In this talk I will discuss invariants of tuples of matrices under two different group actions:

1. Simultaneous Conjugation
2. Left-Right Action

For the entire talk, let K be an algebraically closed field of characteristic 0, and let d and n be positive integers.

1 Simultaneous Conjugation

Consider the action of $SL_n(K)$ on d-tuples of matrices given by

$$Z \cdot (X_1, \dots, X_d) \mapsto (Z^{-1}X_1Z, \dots, Z^{-1}X_dZ),$$

for $Z \in SL_n(K)$ and $X_1, \dots, X_d \in M_{n\times n}(K)$.

Each matrix gives n^2 variables, so we have $m = d \cdot n^2$ total variables.

Question 1. Which polynomials in the $d \cdot n^2$ variables are invariant under this action?

Pop Quiz: Can you name some invariants? (trace, determinant...)

Claim 2. The following collection of polynomials are invariants:

$$\{tr(X_{i_1} \cdot \dots \cdot X_{i_t}) \mid i_j \in [d]\}$$

Proof.

$$\begin{aligned}\mathrm{tr}(A \cdot (X_{i_1} \cdot \dots \cdot X_{i_t})) &= \mathrm{tr}(A^{-1} X_{i_1} A \cdots A^{-1} X_{i_t} A) \\ &= \mathrm{tr}(A^{-1} X_{i_1} \cdot \dots \cdot X_{i_t} A) \\ &= \mathrm{tr}(X_{i_1} \cdot \dots \cdot X_{i_t})\end{aligned}$$

Because $\mathrm{tr}(AB) = \mathrm{tr}(BA)$ □

Proposition 3 (Procesi, Formanek, Razmyslov, Donkin)**.** *The polynomials*

$$\{\mathrm{tr}(X_{i_1} \cdot \dots \cdot X_{i_t}) \mid t \leq n^2,\ i_j \in [d]\}$$

generate the ring of invariants (when K is algebraically closed and characteristic 0).

Example 4. Let $d = 1$, $n = 2$. Then the generators are

$$\mathrm{tr}\begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad \mathrm{tr}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}^2\right) \quad \text{... and 2 more}$$

$$= a + d \qquad = a^2 + 2bc + d^2$$

Since the determinant is an invariant, we should be able to write

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$$

in terms of these. We see that

$$ad - bc = \frac{1}{2}\left((a+d)^2 - (a^2 + 2bc + d^2)\right)$$

Pros: these polynomials are explicit, have low degrees, and can be computed easily.
Cons: the generating set has size $(d+1)^{n^2}$, which is exponential in n.

By applications of Noether Normalization the size of the generating invariants **can** be much smaller in principle.

Question 5. Is there a more efficient way to obtain a generating set?

Theorem 6 (Mulmuley, Forbes - Shpilka)**.** *There is a deterministic* ***polynomial time*** *algorithm to solve the following problem: given two tuples of rational matrices, determine if the closure of their orbits under simultaneous conjugation intersect.*

Proof. The proof idea is in two steps:

1. (Mulmuley) Give a probabilistic polynomial time algorithm by obtaining sufficiently random linear combinations of the invariants by using only polynomially many random bits.

2. (Forbes - Shpilka) De-randomize.

□

2 Left-Right Action

We now consider the following action by $SL_n(K) \times SL_n(K)$ given by

$$(Z, W) \cdot (X_1, \ldots, X_d) \mapsto (Z^{-1}X_1W, \ldots Z^{-1}X_dW)$$

for $(Z, W) \in SL_n(K)$ and $X_1, \ldots, X_d \in M_{n\times n}(K)$.

Again, we have n^2 variables in each matrix, for a total of $d \cdot n^2$ variables.

Question 7. Which polynomials in the $d \cdot n^2$ variables are invariant under this action?

Pop quiz: What are some invariants under this action? (det...)

Proposition 8. *(Many people over several papers) The following collection of polynomials generate the invariants:*

$$\{\det(C_1 \otimes X_1 + \cdots + C_d \otimes X_d) \mid C_i \in M_k(K),\ k \in \mathbb{N}\}.$$

Here, $C \otimes X$ makes an $nk \times nk$ block matrix with blocks $c_{ij}X$.
Pros: concisely described
Cons: infinite

There exits some bound on k, the dimension of the matrix coefficients C_i. After many improvements, Derksen and Makam proved that $k \leq n^2$.

Theorem 9. *(Many people, source gives no single reference) There is a probabilistic polynomial time algorithm to solve the following problem. Given two tuples of rational matrices $(A_1, \ldots, A_d)$, $(B_1, \ldots, B_d)$ to determine if the closure of their orbits under the left-right action intersect. In the case that all B_i are 0, there is a deterministic algorithm.*