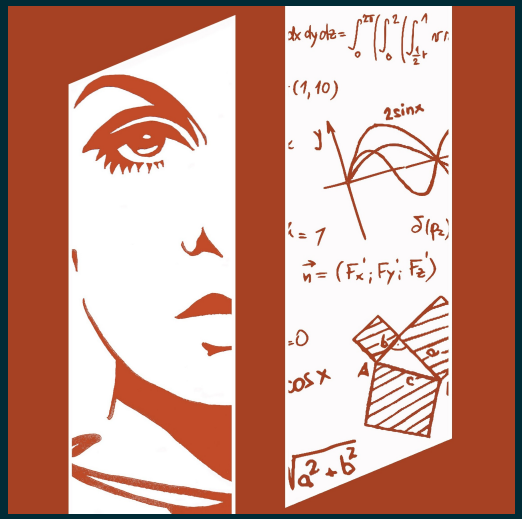
Machine learning and the Iterated Prisoner's Dilemma

SECOND YEAR REPORT

Dr. Vincent Knight Dr. Jonathan Gillard



PLOS Channels

soapbox science

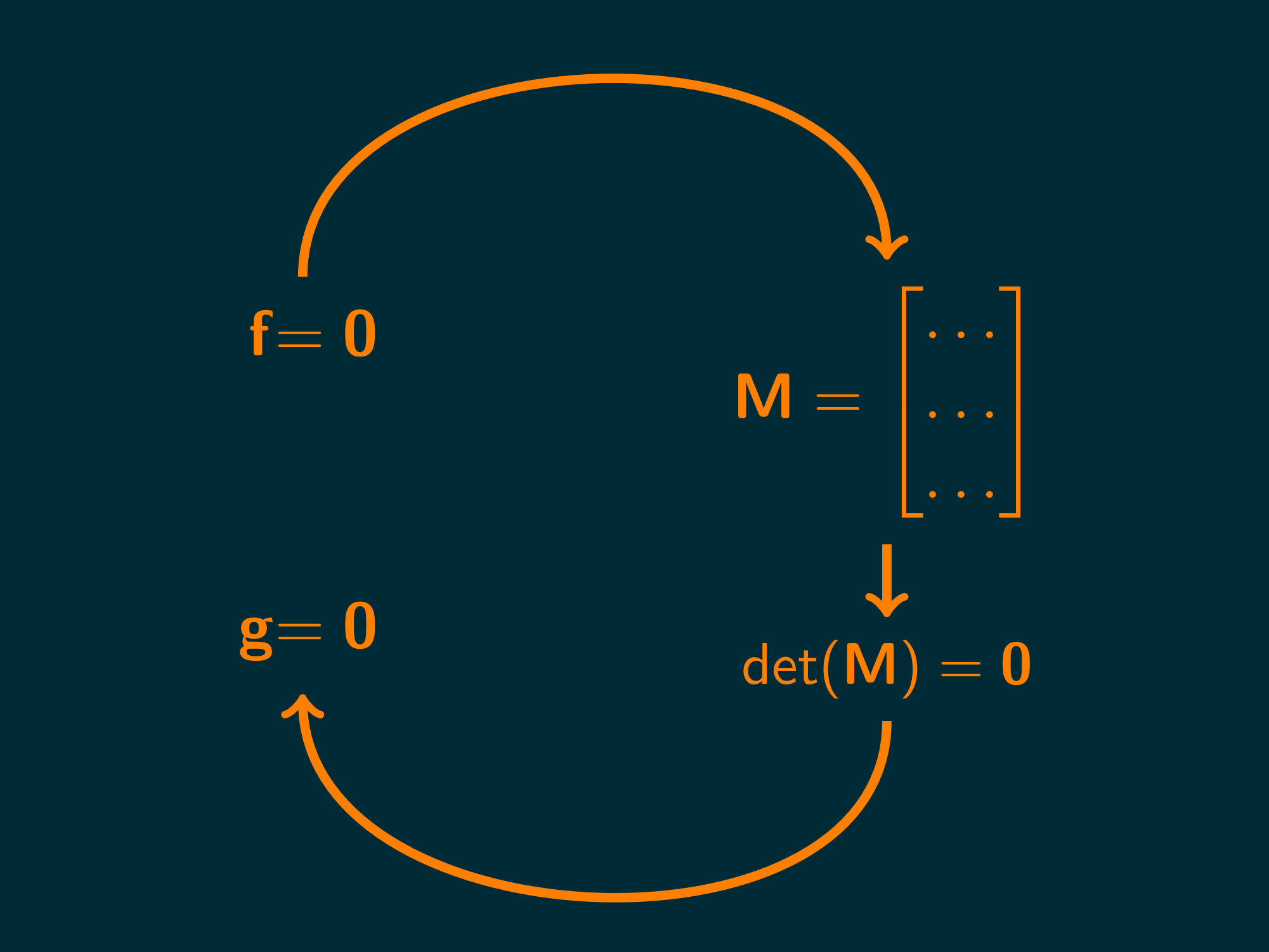
3MT®

FOUNDED BY THE UNIVERSI

- ☑ **Reinforcement Learning Produces Dominant Strategies for the Iterated Prisoner's Dilemma**. PLOS One. 2017.
- ☐ **An Evolutionary Game Theoretic Model of Rhino Horn Devaluation**. Arxiv pre print. 2017. (under review with Ecological Modelling)
- ☐ **Evolution Reinforces Cooperation with the Emergence of Self-Recognition Mechanisms: an empirical study of the Moran process for the iterated Prisoner's dilemma**. Vincent Knight, Arxiv pre print. 2017. (under review with PLOS One)

- ☑ Machine Learning. Duration: **88 hours**.
- ☑ MA3600 Game Theory. Duration: **40 hours**.
- ☑ NATCOR Stochastic Modelling. Duration: **40 hours**.
- ☑ NATCOR Simulation. Duration: **40 hours**.
- ☑ NATCOR Combinatorial Optimisation. Duration: **40 hours**.

Resultant Theory


$f = 0$
$M = \begin{bmatrix} \cdots \\ \cdots \\ \cdots \end{bmatrix}$
$g = 0$
$\det(M) = 0$

$$f = x^2 - 5x + 6$$
$$g = x^2 - 3x + 2$$

$$f = x^2 - 5x + 6$$
$$g = x^2 - 3x + 2$$

$$S = \begin{bmatrix} 1 & -5 & 6 & 0 \\ 0 & 1 & -5 & 6 \\ 1 & -3 & 2 & 0 \\ 0 & 1 & -3 & 2 \end{bmatrix}, \text{ where } det(S) = 0$$

$$f = x^2 + xy + 2x + y - 1$$
$$g = x^2 + 3x - y^2 + 2y - 1$$

$$f = x^2 + xy + 2x + y - 1$$
$$g = x^2 + 3x - y^2 + 2y - 1$$

$$S = \begin{bmatrix} x+1 & x^2+2x-1 & 0 \\ 0 & x+1 & x^2+2x-1 \\ -1 & 2 & x^2+3x-1 \end{bmatrix}$$

$$f = x^2 + xy + 2x + y - 1$$
$$g = x^2 + 3x - y^2 + 2y - 1$$

$$S = \begin{bmatrix} x+1 & x^2+2x-1 & 0 \\ 0 & x+1 & x^2+2x-1 \\ -1 & 2 & x^2+3x-1 \end{bmatrix}$$

$$det(S) = -x\,(x-1)\,(x+3)$$

Sylvester

Bezout

Macaulay

Dixon

sympy / sympy

Unwatch 316 Star 4,837 Fork 2,285

Code Issues 2,763 Pull requests 560 Projects 0 Wiki Insights

implementation of multivariate resultants. #14370

Edit

Merged **jksuom** merged 19 commits into `sympy:master` from `Nikoleta-v3:multivariate-resultants` on 30 Mar

Conversation 63 Commits 19 Checks 0 Files changed 6 +3,033 −0

Nikoleta-v3 commented on 2 Mar — Contributor

Brief description

Adds a new file `multivariate_resultants` that contains two classes.
These classes are implementations of the following multivariate resultants:

- Dixons
- Macaulay

They are a natural follow up from the resultants implemented withing the library:

- subresultants_qq_zz.py

Resultants are used to indentify if polynomials have common roots. The multivariate version is for multivariate systems.

Other comments

Tests have been implemented and are currently passing. I added relevant literature and some notebooks with examples in the example/notebooks/ directory.

References to other Issues or PRs

Reviewers

asmeurer
smichr
jksuom
normalhuman

Assignees

No one assigned

Labels

PR: sympy's turn

Projects

None yet

Milestone

No milestone

Thank you!