Can Explainable AI Give Insights for Symbolic Computation?

Matthew England (Coventry University, UK)

Special Session in honour of James H. Davenport
SYNASC 2023
Nancy, France 12th September 2023

Author currently supported by EPSRC grant EP/T015748/1.

Personal Note

First met JHD interviewing for a PDRA position with him in late 2011.

Worked with JHD in Bath for three years: 2012–2015.

Continued to collaborate after move to Coventry in 2015, e.g. EU SC-Square Project and now the EPSRC DEWCAD Project.

Co-authored at least one paper with James every year since 2012! Always a pleasure.

Happy Birthday James!

Outline

Key Reference: L. Pickering, T. Del Rio Almajano, M. England and K. Cohen. *Explainable AI Insights for Symbolic Computation: A case study on selecting the variable ordering for cylindrical algebraic decomposition.* Submitted, 2023.

Preprint: https://arxiv.org/abs/2304.12154

Symbolic Computation VS Machine Learning (Slide 2/18)

Symbolic Computation refers to algorithms and software for manipulating exact mathematical expressions and objects.

Machine Learning (ML) can use statistics and big data to learn how to perform tasks that have not been explicitly programmed.

(Q) Can ML replace symbolic computation?

There is a growing body of research on the use of ML in place of expensive symbolic computation. E.g. for symbolic integration and the solution of differential equations [Lample and Charton, 2020].

There are over-fitting issues: where the ML does very well on the data used to train it but poorly on different data. It is not always obvious which data is similar to the training data.

For [Lample and Charton, 2020] it is cheap to symbolically check the correctness; not so for most symbolic computation.

Symbolic Computation WITH Machine Learning (Slide 3/18)

ML can only offer probabilistic guidance, but symbolic computation prizes exact results. 99% accuracy is great for image recognition but would not be acceptable for a mathematical proof.

However: ML can be applied to symbolic computation and still ensure exact results; by having it guide existing algorithms rather than replace them entirely.

Computer Algebra algorithms will often come with choices that need to be made but which do not effect the mathematical correctness of the final result; but do effect the resources required to find that result, and how the result is presented.

Such choices are often either left to the user, hard coded by the developer, or made based on a simple heuristic. ML can offer a superior choice.

ML to Optimise Computer Algebra Examples (Slide 4/18)

- Huang et al. [2014] was the first use of ML for computer algebra: used to select the CAD variable ordering (our topic).
- Kuipers et al. [2015] used a Monte-Carlo tree search to find the representation of polynomials that are most efficient to evaluate numerically.
- Simpson et al. [2016] used ML to choose which algorithm to compute the resultant with for a given problem instance.
- Brown and Daves [2020] used a neural network to select the order of polynomial constraints to process in their solver.
- Peifer et al. [2020] applied reinforcement learning to select which S-Pair to process next when building a Gröbner Basis.

Fuller survey in [Pickering et al., 2023].

This is INTERESTING Machine Learning

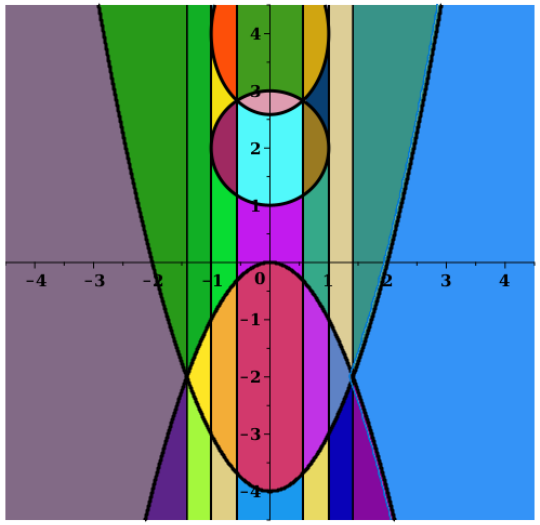
So ML has great potential for Symbolic Computation. But note this is also a particularly challenging / interesting ML domain:

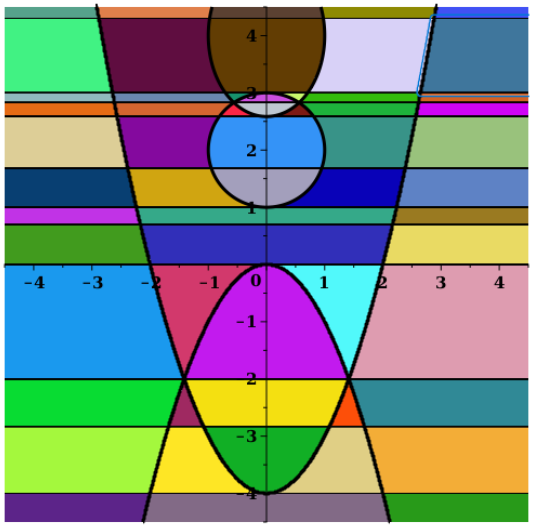
- No a priori limit on the input space.
- Supervised learning hard: because labelling dataset needs lots of expensive symbolic computation.
- Unsupervised learning is hard: because it is unclear if a particular outcome is good or bad without seeing the competition!
- What constitutes a meaningful and representative data set?
- Insufficient quantities of real world data for deep learning.
- How to perform synthetic data generation to allow for good generalisability on problems of interest?

Case Study: Variable Ordering for CAD

Cylindrical Algebraic Decomposition (CAD) [Collins, 1975] is a key tool for semi-algebraic sets / formulae in non-linear real arithmetic. Requires a variable ordering, whose choice affects computation time, even complexity Brown and Davenport [2007].

Some human-designed heuristics for the choice use only simple metrics upon the input [Brown, 2004], [del Río and England, 2022]; while others make use of more expensive algebraic information [Dolzmann et al., 2004], [Bradford et al., 2013], [England et al., 2014], [Wilson et al., 2014].





ML for CAD Variable Ordering I

Huang et al. [2014] was the first use of ML for computer algebra: a support vector machine was trained to choose which of our three human-made heuristics to follow when selecting the CAD variable ordering.

We observed subsets of problems on which each heuristic was dominant. The ML meta-heuristic was better than those of any one human-made heuristic.

We later applied this methodology to other CAD algorithm choices, e.g Gröbner Basis preconditioning Huang et al. [2019].

ML for CAD Variable Ordering II

Work continued with more involved ML tools for the problem:

- Variety of classification models [England and Florescu, 2019];
- Automated feature generation technique [Florescu and England, 2019];
- Partial training based on runtime instead of accuracy [Florescu and England, 2020a];
- An open source software pipeline [Florescu and England, 2020b].

Also in this period Chen et al. [2020] experimented with a combination of a greedy heuristic to identify candidate solutions and neural networks to choose between them.

QFNRA Dataset Lessons in del Rio and England [2023]

Work has focussed on the QFNRA benchmark set in the SMT-LIB (the most substantial set of problems admissible to CAD). But:

- Data is unbalanced with respect to variable ordering. This will lead to overfitting to the most prevalent ordering.

Addressed by permuting variable labels. Additional data generation this way (augmentation) gives genuine learning that makes up for the unfair advantage lost by balancing.

- The benchmarks contain many almost identical problems. Lead to data leakage between training and testing sets.

Addressed by merging problems instances who have the same CAD tree structure for all orderings. This reduces the dataset to a sixth but with the same learning achieved.

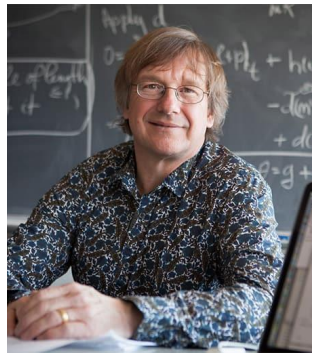
Outline

Peifer, Stillman, and Halpern-Leistner

Peifer et al. [2020] applied ML to choose the order in which to process S-pairs in Buchberger's algorithm for a Gröbner Basis. Their model outperformed human-made heuristics for the choice.

A human analysis of their model revealed preferences for pairs whose S-polynomials are low degree and those which are monomials. Basing the decision on the S-polynomials rather than S-pairs was novel.

Hand made heuristics based on these insights outperformed the prior human-made heuristics (but not the full ML model).



Beyond Efficiency Gains

The work of Peifer et al. [2020] suggests that ML may be able to offer Computer Algebra something beyond efficiency gains: new ideas to explore and human-level heuristics.

(Working) Definition: A "*human-level*" strategy for making a heuristic decision is one that may be described clearly in natural language in a quantity of text of the same order or magnitude as existing heuristics created by humans.

Why prefer human-level heuristics?

- Easier to understand.
- Easier to include in an implementation.
- Less risk of over-fitting to a dataset.

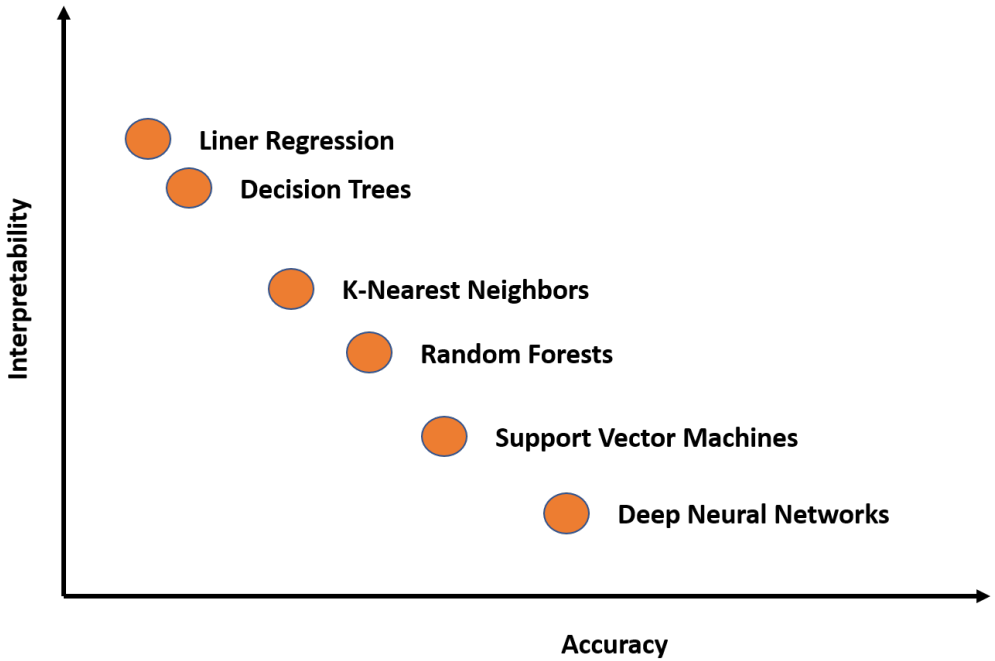
We suggest the creation of human-level heuristics through the use of XAI techniques.

What is XAI?

XAI (*eXplainable AI*) is an emerging domain. It encompasses both AI methods that are inherently interpretable to human experts, and tools to analyse less interpretable AI to produce explanations for their behaviour.

Note that there is a perceived trade-off between ML accuracy and interpretability.

However, this trade-off has been shown to differ depending on the problem domain and user!



Recent Work on XAI for CAD Variable Ordering (Slide 13/18)

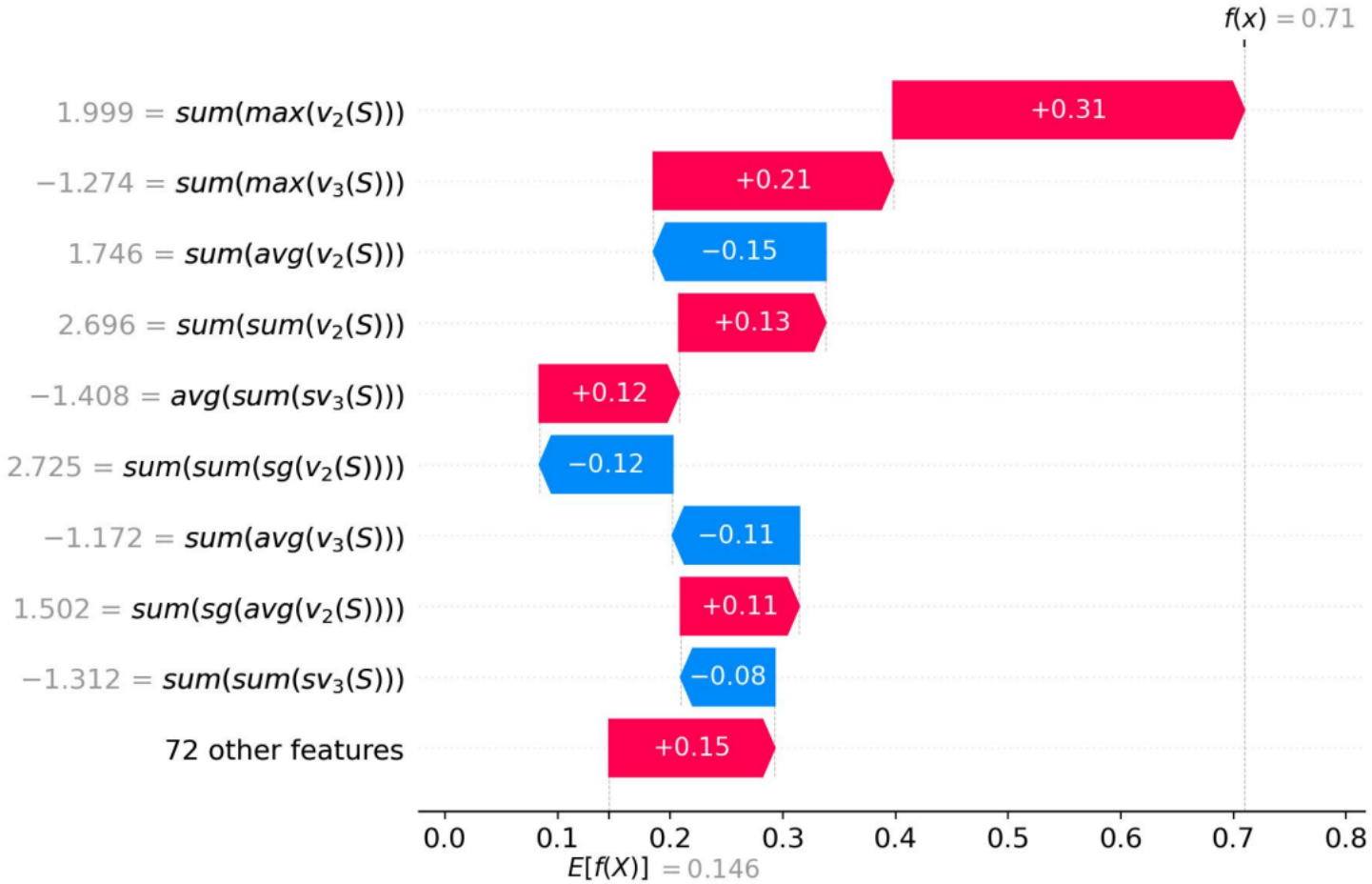
Joint with Lynn Pickering and Tereso del Rio.

We worked with the software pipeline of Florescu and England [2020a]. This represents each CAD instance as a vector of floating points defined by (simple) features of the input polynomials

We used the popular XAI tool, SHAP [Lundberg and Lee, 2017], which identifies the relative influence of each feature to a model's prediction. Based on Shapley Values in game theory where we play a game with subsets of players to identify the contribution of each: here players are ML features.

SHAP Waterfall Plot Example

Produced to explain one model's prediction on one instance for a single variable ordering.



Our Methodology

We ran an experiment with the 3-variable problems from the QFNRA benchmarks in the SMT-LIB (with the balancing and merging discussed earlier).

- Ran SHAP for each four ML models (KNN, DT, SVM, MLP).
- Combined scores across instances.
- Merged features of same metric on different variables.
- Applied a Borda Count vote to combine the feature rankings of the four models into a single ranking.
- Formed human-level heuristics from triples of the best ranked features.
  - The triples query one feature, breaking ties with the next.
  - They are applied greedily: pick first variable based on input, then project one dimension and pick second variable based on that projection, etc.

Results – Final Feature Ranking

Feature Name	Voted Score	
$sum(max(v_i(S)))$	3.333	mods
$avg(avg(v_i(S)))$	2.167	
$sum(sum(v_i(S)))$	1.158	
$avg(avg(sg(v_i(S))))$	1.15	
$sum(sg(sum(v_i(S))))$	0.794	
$sum(max(sv_i(S)))$	0.787	
$avg(avg(sv_i(S)))$	0.583	
$sum(sum(sg(v_i(S))))$	0.554	Brown3
$max(sum(v_i(S)))$	0.475	
$sum(sum(sv_i(S)))$	0.472	
$max(max(v_i(S)))$	0.467	Brown1
$max(sum(sg(v_i(S))))$	0.3	
$max(avg(v_i(S)))$	0.245	
$max(max(sg(v_i(S))))$	0.218	
$max(avg(sg(v_i(S))))$	0.21	
$max(sum(sv_i(S)))$	0.209	
$max(avg(sv_i(S)))$	0.192	
$max(max(sv_i(S)))$	0.191	Brown2

XAI identified as top the feature corresponding to current state-of-the-art in [del Río and England, 2022].

Experimented with all 120 ordered triples you can form by taking three from the top six in the ranking.

Results – Heuristics

Name	Accuracy	Total time	Markup	# Completed
gmods	0.563	**7192.2**	**0.212**	**982.6**
Brown	0.553	7842.6	0.278	968.9
mods	**0.639**	8137	0.127	979
random	0.167	20797.3	4.034	262.5
free-mods	0.639	6637	0.566	990
virtual-best	1	4822.7	0	1019

Name	Accuracy	Total time	Markup	# Completed
SumMaxV	**0.563**	7192.2	**0.212**	982.6
AvgAvgV	0.544	**7138.7**	0.224	**983.5**
SumSumV	0.549	7524.8	0.261	975.3
AvgAvgSgV	0.535	8682.6	0.559	956.3
SumSgSumV	0.45	10836.7	1.223	922.5
SumMaxSV	0.509	8771.7	0.563	956.5

Name	Accuracy	Total time	Markup	# Completed
Brown	0.553	7842.6	0.278	968.9
T1	0.567	**6896.3**	0.193	**985.7**
T2	**0.583**	6896.7	**0.188**	984.8

Results Analysis

- Both `T1` and `T2` (the best performing of the 120 triples) could be seen as the new state-of-the-art human level heuristic for the problem (which depends on preferred evaluation metric).
- Either may be simply encoded into any CAD implementation without recourse to AI software.
- The features XAI ranked most important do well on their own too: one is the key metric in the prior-state-of-the-art but the other had not been studied before.
- `T1` was in fact formed from the the top three XAI ranked features in that order.

Conclusion: XAI may be used to produce human-level heuristics for computer algebra.

Contact Details

Contact Details

Matthew.England@coventry.ac.uk

https://matthewengland.coventry.domains/

Thanks for Listening

Bibliography I

R. Bradford, J. H. Davenport, M. England, and D. Wilson. Optimising problem formulations for cylindrical algebraic decomposition. In J. Carette, D. Aspinall, C. Lange, P. Sojka, and W. Windsteiger, editors, *Intelligent Computer Mathematics*, volume 7961 of *Lecture Notes in Computer Science*, pages 19–34. Springer Berlin Heidelberg, 2013. URL http://dx.doi.org/10.1007/978-3-642-39320-4_2.

C. W. Brown. Companion to the tutorial: Cylindrical algebraic decomposition, presented at ISSAC '04. URL http://www.usna.edu/Users/cs/wcbrown/research/ISSAC04/handout.pdf, 2004.

C. W. Brown and J. H. Davenport. The complexity of quantifier elimination and cylindrical algebraic decomposition. In *Proceedings of the 2007 International Symposium on Symbolic and Algebraic Computation*, ISSAC '07, pages 54–60. ACM, 2007. URL https://doi.org/10.1145/1277548.1277557.

Bibliography II

C. W. Brown and G. C. Daves. Applying machine learning to heuristics for real polynomial constraint solving. In A. Bigatti, J. Carette, J. H. Davenport, M. Joswig, and T. de Wolff, editors, *Mathematical Software – ICMS 2020*, volume 12097 of *Lecture Notes in Computer Science*, pages 292–301. Springer International Publishing, 2020. URL https://doi.org/10.1007/978-3-030-52200-1_29.

B. Caviness and J. Johnson. *Quantifier Elimination and Cylindrical Algebraic Decomposition*. Texts & Monographs in Symbolic Computation. Springer-Verlag, 1998. URL https://doi.org/10.1007/978-3-7091-9459-1.

Bibliography III

C. Chen, Z. Zhu, and H. Chi. Variable ordering selection for cylindrical algebraic decomposition with artificial neural networks. In A. Bigatti, J. Carette, J. H. Davenport, M. Joswig, and T. de Wolff, editors, *Mathematical Software – ICMS 2020*, volume 12097 of *Lecture Notes in Computer Science*, pages 281–291. Springer International Publishing, 2020. URL https://doi.org/10.1007/978-3-030-52200-1_28.

G. E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Proceedings of the 2nd GI Conference on Automata Theory and Formal Languages*, pages 134–183. Springer-Verlag (reprinted in the collection Caviness and Johnson [1998]), 1975. URL https://doi.org/10.1007/3-540-07407-4_17.

Bibliography IV

T. del Rio and M. England. Data augmentation for mathematical objects. In E. Ábrahám and T. Sturm, editors, *Proceedings of the 8th Workshop on Satisfiability Checking and Symbolic Computation (SC^2 2023)*, number 3455 in CEUR Workshop Proceedings, pages 29–38, 2023. URL http://ceur-ws.org/Vol-3455/.

Tereso del Río and Matthew England. New heuristic to choose a cylindrical algebraic decomposition variable ordering motivated by complexity analysis. In François Boulier, Matthew England, Timur M. Sadykov, and Evgenii V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing*, volume 13366 of *Lecture Notes in Computer Science*, pages 300–317. Springer International Publishing, 2022. URL https://doi.org/10.1007/978-3-031-14788-3_17.

Bibliography V

A. Dolzmann, A. Seidl, and T. Sturm. Efficient projection orders for CAD. In *Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*, ISSAC '04, pages 111–118. ACM, 2004. URL https://doi.org/10.1145/1005285.1005303.

M. England and D. Florescu. Comparing machine learning models to choose the variable ordering for cylindrical algebraic decomposition. In C. Kaliszyk, E. Brady, A. Kohlhase, and C. C. Sacerdoti, editors, *Intelligent Computer Mathematics*, volume 11617 of *Lecture Notes in Computer Science*, pages 93–108. Springer International Publishing, 2019. URL https://doi.org/10.1007/978-3-030-23250-4_7.

Bibliography VI

M. England, R. Bradford, C. Chen, J. H. Davenport, M. Moreno Maza, and D. Wilson. Problem formulation for truth-table invariant cylindrical algebraic decomposition by incremental triangular decomposition. In S. M. Watt, J. H. Davenport, A. P. Sexton, P. Sojka, and J. Urban, editors, *Intelligent Computer Mathematics*, volume 8543 of *Lecture Notes in Artificial Intelligence*, pages 45–60. Springer International, 2014. URL http://dx.doi.org/10.1007/978-3-319-08434-3_5.

D. Florescu and M. England. Algorithmically generating new algebraic features of polynomial systems for machine learning. In J. Abbott and A. Griggio, editors, *Proceedings of the 4th Workshop on Satisfiability Checking and Symbolic Computation (SC^2 2019)*, number 2460 in CEUR Workshop Proceedings, 2019. URL http://ceur-ws.org/Vol-2460/.

Bibliography VII

D. Florescu and M. England. Improved cross-validation for classifiers that make algorithmic choices to minimise runtime without compromising output correctness. In D. Slamanig, E. Tsigaridas, and Z. Zafeirakopoulos, editors, *Mathematical Aspects of Computer and Information Sciences (Proc. MACIS '19)*, volume 11989 of *Lecture Notes in Computer Science*, pages 341–356. Springer International Publishing, 2020a. URL https://doi.org/10.1007/978-3-030-43120-4_27.

D. Florescu and M. England. A machine learning based software pipeline to pick the variable ordering for algorithms with polynomial inputs. In A. Bigatti, J. Carette, J. H. Davenport, M. Joswig, and T. de Wolff, editors, *Mathematical Software – ICMS 2020*, volume 12097 of *Lecture Notes in Computer Science*, pages 302–322. Springer International Publishing, 2020b. URL https://doi.org/10.1007/978-3-030-52200-1_30.

Bibliography VIII

Z. Huang, M. England, D. Wilson, J. H. Davenport, L. Paulson, and J. Bridge. Applying machine learning to the problem of choosing a heuristic to select the variable ordering for cylindrical algebraic decomposition. In S. M. Watt, J. H. Davenport, A. P. Sexton, P. Sojka, and J. Urban, editors, *Intelligent Computer Mathematics*, volume 8543 of *Lecture Notes in Artificial Intelligence*, pages 92–107. Springer International, 2014. URL http://dx.doi.org/10.1007/978-3-319-08434-3_8.

Z. Huang, M. England, D. Wilson, J. Bridge, J. H. Davenport, and L. Paulson. Using machine learning to improve cylindrical algebraic decomposition. *Mathematics in Computer Science*, 13 (4):461–488, 2019. URL https://doi.org/10.1007/s11786-019-00394-8.

Bibliography IX

J. Kuipers, T. Ueda, and J. A. M. Vermaseren. Code optimization in FORM. *Computer Physics Communications*, 189:1–19, 2015. URL https://doi.org/10.1016/j.cpc.2014.08.008.

G. Lample and D. Charton. Deep learning for symbolic mathematics. In S. Mohamed, M. White, K. Cho, and D. Song, editors, *Eighth International Conference on Learning Representations (ICLR 2020)*, 2020. URL https://iclr.cc/virtual_2020/poster_S1eZYeHFDS.html.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017)*, pages 4768–4777. Curran Associates Inc., 2017. URL https://dl.acm.org/doi/10.5555/3295222.3295230.

Bibliography X

D. Peifer, M. Stillman, and D. Halpern-Leistner. Learning selection strategies in Buchberger's algorithm. In H. Daumé III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, volume 119 of *Proceedings of Machine Learning Research*, pages 7575–7585. PMLR, 2020. URL https://proceedings.mlr.press/v119/peifer20a.html.

Lynn Pickering, Tereso Del Rio Almajano, Matthew England, and Kelly Cohen. Explainable AI insights for symbolic computation: A case study on selecting the variable ordering for cylindrical algebraic decomposition. *Submitted*, 2023. URL https://arxiv.org/abs/2304.12154.

Bibliography XI

Matthew C. Simpson, Qing Yi, and Jugal Kalita. Automatic algorithm selection in computational software using machine learning. In *15th IEEE International Conference on Machine Learning and Applications (ICMLA 2016)*, pages 355–360, 2016. URL https://doi.org/10.1109/ICMLA.2016.0064.

D. Wilson, M. England, J. H. Davenport, and R. Bradford. Using the distribution of cells by dimension in a cylindrical algebraic decomposition. In *16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, SYNASC '14, pages 53–60. IEEE, 2014. URL http://dx.doi.org/10.1109/SYNASC.2014.15.