The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

The Automatic Computer Scientist

Andrew Cropper

University of Oxford
andrew.cropper@cs.ox.ac.uk

Algorithms are ubiquitous: they track our sleep, help us find cheap flights, and even help us see black holes. However, designing novel algorithms is extremely difficult, and we do not have efficient algorithms for many fundamental problems. **The goal of my research is to accelerate algorithm discovery by building an *automatic computer scientist*.** To work towards this goal, my research focuses on inductive logic programming (ILP) (Cropper and Dumancic 2022), a form of machine learning (ML) in which my collaborators and I have demonstrated major advances in automated algorithm discovery over the past five years (Cropper et al. 2022).

Metagol

My PhD research focused on Metagol (Cropper and Muggleton 2016), which overcomes three long-standing grand challenges: learning recursive programs, learning efficient programs, and predicate invention.

Recursion. Recursion is essential to represent algorithms with loops and thus algorithms that generalise. Learning recursive programs has been a grand challenge. Metagol overcomes this challenge by using *metarules* to guide the search (Cropper and Tourret 2020). Metagol was the first system that could learn a wide variety of recursive programs, such as robot strategies, sorting algorithms, and string transformation programs.

Efficiency. When learning programs, we ideally want to learn efficient programs. However, existing techniques could not distinguish between the efficiencies of programs, such as permutation sort (O(n!)) and merge sort (O(n log n)). To address this limitation, I introduced a technique that is guaranteed to learn the most efficient program for a given problem (Cropper and Muggleton 2019).

Predicate invention. Predicate invention is the automatic invention of new high-level reusable concepts. The ability to invent new predicate symbols is essential to learn solutions for many complex problems. For instance, when learning to solve programming puzzles, Metagol invents reusable list operations, such as *reverse* and *droplast* (Cropper, Morel, and Muggleton 2020).

Popper

To improve the scalability of ILP approaches, we introduced an approach called *learning from failures* (LFF) (Cropper and Morel 2021; Cropper 2022; Hocquette and Cropper 2022). LFF frames the ILP problem as a constraint satisfaction problem (CSP), where each solution to the CSP represents a hypothesis. The goal of a LFF learner is to accumulate constraints to eliminate solutions and thus hypotheses. A learner does so by continually building hypotheses and testing them on examples. If a hypothesis is incorrect, we explain why. We then use the explanation to build constraints to add to the CSP. For instance, if a program is too general (entails a negative example), the constraints prune its generalisations. If a program is too specific (entails a negative example), the constraints prune its specialisations. We have shown that POPPER, an LFF system, can learn programs 10 times larger and 100 times faster than existing approaches.

References

Cropper, A. 2022. Learning Logic Programs Though Divide, Constrain, and Conquer. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*, 6446–6453. AAAI Press.

Cropper, A.; and Dumancic, S. 2022. Inductive Logic Programming At 30: A New Introduction. *J. Artif. Intell. Res.*, 74: 765–850.

Cropper, A.; Dumancic, S.; Evans, R.; and Muggleton, S. H. 2022. Inductive logic programming at 30. *Mach. Learn.*, 111(1): 147–172.

Cropper, A.; and Morel, R. 2021. Learning programs by learning from failures. *Mach. Learn.*, (4): 801–856.

Cropper, A.; Morel, R.; and Muggleton, S. 2020. Learning higher-order logic programs. *Machine Learning*, (7): 1289–1322.

Cropper, A.; and Muggleton, S. H. 2016. Metagol System. https://github.com/metagol/metagol.

Cropper, A.; and Muggleton, S. H. 2019. Learning efficient logic programs. *Mach. Learn.*, (7): 1063–1083.

Cropper, A.; and Tourret, S. 2020. Logical reduction of metarules. *Mach. Learn.*, (7): 1323–1369.

Hocquette, C.; and Cropper, A. 2022. Learning programs with magic values. *Mach. Learn.*