Machine Learning for Theory Exploration

Chris Warburton
University of Dundee
cmwarburton@dundee.ac.uk

Ekaterina Komendantskaya
University of Dundee
katya@dundee.ac.uk

ABSTRACT

Theory Exploration (TE) aims to automate the tedious yet necessary task of verifying Mathematicians' and programmers' work, but the blind search used by existing approaches limits them to small examples. Meanwhile, huge repositories of formal knowledge are being routinely data-mined for structure and correlation. We provide a method for guiding TE using these abundant statistics, and assess whether this hybrid approach is feasible for tackling problems of a realistic size.

INTRODUCTION

Small mistakes made in Mathematics and programming can cause large problems in the real world. Computer verification can reduce this risk by checking our reasoning step-by-step; but traditional software's inability to follow "obvious" arguments without explicit guidance often makes it impractical. The recent *theory exploration* approach[1] tackles this by generating a database of facts about a user-provided "theory" (eg. a software library). This database can either serve as "background knowledge" for a traditional verification tool, to help it follow more coarse-grained proof steps; or it can be queried directly to discover interesting facts about the theory.

Existing TE systems, such as QuickSpec[3] and HipSpec[2], rely on undirected, brute-force search to generate their databases. Although complete, these algorithms' exponential complexity limits their scalability to small theories with only a handful of definitions. To be practical, the technique must be able to work with theories spanning thousands of definitions, without relying on expert users to cherry-pick a sub-set.

Machine Learning (ML) offers many scalable techniques for studying large theories, such as *premise selection*[5]: choosing lemmas which are most likely to help us prove a conjecture. This is similar to our cherry-picking problem, but TE is concerned with a theory's structure rather than individual conjectures.

Theory structure has been studied by ML4PG[4], using *clustering* to find statistical similarities and hierarchies among definitions. We investigate whether such clustering information can help TE systems navigate large theories more effectively.

METHODOLOGY

Following the approach of QuickSpec, our "theories" are software libraries written in Haskell; a popular programming language where correctness is concerned. We divide these definitions into small clusters, using a similar technique to ML4PG. We then run QuickSpec on each cluster individually, to produce "facts" in the form of equalities relating the definitions. The facts for each cluster are then combined.

We compare the total running time and the resulting database against running QuickSpec on the whole, unclustered library. We observe the tradeoff between speed and completeness, for various library sizes.

RESULTS

We have developed an ML approach for analysing Haskell code, including a bespoke feature extraction method which, to our knowledge, is the first in the literature. The success of our clustering tool shows the suitability of Haskell for the same statistical approaches used to study other languages.

FUTURE WORK

Our tool can be improved with a more systematic consideration of the design decisions and heuristics used, but the preprocessor approach is inherently limited. A more thorough integration of our methodology into state-of-the-art TE systems would provide more opportunities for exploiting statistical information, and also to "close the loop" by using the TE database to inform the statistical algorithms.

References

[1]Buchberger, B. 2000. Theory exploration with theorema. *Analele Universitatii Din Timisoara, ser. Matematica-Informatica.* 38, 2 (2000), 9–32.

[2]Claessen, K. et al. HipSpec: Automating inductive proofs of program properties. *In workshop on automated theory eXploration: ATX 2012.*

[3]Claessen, K. et al. 2010. QuickSpec: Guessing formal specifications using testing. *Tests and proofs.* G. Fraser and A. Gargantini, eds. Springer Berlin Heidelberg. 6–21.

[4]Heras, J. and Komendantskaya, E. 2013. Statistical proof pattern recognition: Automated or interactive? *CoRR.* abs/1303.1419, (2013).

[5]Kühlwein, D. et al. 2012. Overview and evaluation of premise selection techniques for large theory mathematics. *Automated reasoning.* Springer. 378–392.