Bayesian Optimisation for Premise Selection in Automated Theorem Proving (Student Abstract)

Agnieszka Słowik, Chaitanya Mangla, Mateja Jamnik, Sean B. Holden, Lawrence C. Paulson
University of Cambridge, Department of Computer Science and Technology
William Gates Building, 15 JJ Thomson Ave, Cambridge CB3 0FD, UK
agnieszka.slowik@cl.cam.ac.uk

Abstract

Modern theorem provers utilise a wide array of heuristics to control the search space explosion, thereby requiring optimisation of a large set of parameters. An exhaustive search in this multi-dimensional parameter space is intractable in most cases, yet the performance of the provers is highly dependent on the parameter assignment. In this work, we introduce a principled probabilistic framework for heuristic optimisation in theorem provers. We present results using a heuristic for premise selection and the Archive of Formal Proofs (AFP) as a case study.

Introduction

Theorem provers use heuristics at various points in their operation, such as in search control and premise selection. These heuristics often have parameters that greatly influence the practical performance of a prover. Existing approaches to selecting such parameters require human supervision, rules of thumb or extensive testing (Hoder and Voronkov 2011). Such testing is often conducted on large theory sets, and is thus computationally expensive. For instance, Open CYC (Matuszek, Cabral, and Wirbrock 2006) contains over 3 million axioms while each of the problems has a proof involving up to 20 premises. An alternative to the exhaustive search is to sparsely navigate the multi-dimensional space of parameters. We argue that probabilistic search enables efficient and automated optimisation of parameterised heuristics in theorem proving.

Here, we explore Bayesian Optimisation (Močkus 1975) with Gaussian Processes (GPs) (Rasmussen and Williams 2005) as a general solution to efficient heuristics tuning in automated theorem proving. We conduct a case study in premise selection using a state-of-the-art heuristic Sumo Inference Engine (SInE) (Hoder and Voronkov 2011). Our framework based on GPs takes at most nine minutes to find the optimal set of parameters in ten AFP articles. The premises recommended by the optimised SInE were sufficient to prove 85.3% of the conjectures using Sledgehammer (Böhme and Nipkow 2010).

The Sequential Model-based Algorithm Configuration framework (Hutter, Hoos, and Leyton-Brown 2011) takes a similar approach, but is distinguished from our work due to its use of the *Expected Improvement* criterion. We employ the *Upper Confidence Bound* (UCB) criterion, as explained below. The parameter κ available in UCB allows a fine tuning of the exploration-exploitation trade-off.

Method

Premise selection can be defined as follows:

Definition 1 *Given a set of premises $\mathbb{P}$, an Automated Theorem Prover $\mathcal{A}$ and a new conjecture C, select the premises from $\mathbb{P}$ that will most likely lead to a proof of C by $\mathcal{A}$.*

SInE was developed for premise selection in large theories; it filters lemmas based on the symbols used in their statement. Symbols are considered globally rare if their frequency is less than g across all theories; and locally rare if they occur infrequently within the lemma, with a frequency less than t times the frequency of all local symbols. The premise set $\mathbb{P}$ is built inductively, starting with the goal theorem. For every globally or locally rare symbol σ in $\mathbb{P}$, any lemma containing σ is added to the set $\mathbb{P}$, and this is repeated for a maximum of k iterations. The algorithm is therefore parameterised by one continuous parameter t and two discrete parameters g and k. As was demonstrated in (Hoder and Voronkov 2011), these parameters greatly influence the performance of the algorithm.

We maximise an objective function which is correlated to the number of conjectures an ATP system would be able to prove given the premises selected by SInE. We assume this function was sampled from a GP. As we evaluate the performance of SInE given the point in the parameter space, the Bayesian Optimisation framework improves the posterior distribution for the objective function as the agent becomes more certain of which regions are worth exploring. In our implementation we choose the point in the parameter space to be evaluated in the next iteration based on the posterior distribution and upper confidence bound of a GP which is one of the standard methods referred to as the Gaussian Process-Upper Confidence Bound (GP-UCB) algorithm (Srinivas et al. 2010)

Table 1: Premise selection results on the AFP articles.

AFP article	Nr of goals	Proofs found [%]	Time [s]	Optimal parameters
Polynomials	135	87%	57s	t: 16.3, g: 58, k: 131
AbstractHoareLogics	793	63%	249s	t: 17.6, g: 57, k: 130
Completeness	475	89%	151s	t: 18.9, g: 63, k: 134
FinFun	263	95%	73s	t: 19.6, g: 57, k: 132
HeardOf	716	93%	331s	t: 19.5, g: 57, k: 131
InductiveConfidentiality	1425	82%	451s	t: 19.6, g: 58, k: 130
RefineMonadic	1509	95%	522s	t: 14.7, g: 64, k: 123
MiniML	345	84%	104s	t: 19.1, g: 58, k: 131
RecursionTheory	656	85%	205s	t: 19, g: 57, k: 130
SortEncodings	776	80%	437s	t: 14, g: 64, k: 123

Experiments

Dataset AFP (Jaskelioff and Merz 2005) is a collection of proofs formalised in Isabelle (Nipkow, Wenzel, and Paulson 2002). We used a parsed version of the dataset that meets the input requirements of MaSh (Kühlwein et al. 2013), the machine learning premise selector currently implemented in Isabelle. Here, we report the results on 10 articles containing various theories and of sizes ranging from around 100 to around 1500 conjectures. Each conjecture was paired with a history of premises extracted from Sledgehammer logs that were used to determine which lemmas are needed to prove a goal.

Evaluation In premise selection, it is acceptable to provide more premises than necessary to prove a conjecture in order to minimise the risk of missing a key lemma. However, the main purpose of filtering is to lower the cost of considering irrelevant lemmas, and so an efficient algorithm should minimise the number of unnecessary recommendations. To let this trade-off guide the optimisation process, we use a metric based on precision and recall . At the testing stage we evaluate the algorithm based on the number of conjectures that would be proved in practice by Sledgehammer using the premises recommended by SInE. We assume that all of the premises used by Sledgehammer are necessary to prove the conjecture whereas in practice the prover might be able to find an alternative solution that requires a different set of premises. Consequently, this testing metric will tend to underestimate the number of conjectures proved using the SInE recommendations.

Analysis Preliminary results (see Table 1) suggest that the framework is efficient in finding the optimal parameter combination across different theories. This allows us to explore a wider range of parameters and produce an offline heuristic recommendation to a theorem prover.

Future Work

A possible future direction involves reproducing our premise selection experiments on a larger set of conjectures, and applying the Bayesian Optimisation framework to another bottleneck in automated theorem proving, for example strategy scheduling.

References

Böhme, S., and Nipkow, T. 2010. Sledgehammer: Judgement day. In *Proceedings of the 5th International Conference on Automated Reasoning*, 107–121. Berlin, Heidelberg: Springer-Verlag.

Hoder, K., and Voronkov, A. 2011. Sine qua non for large theory reasoning. In Bjørner, N., and Sofronie-Stokkermans, V., eds., *Automated Deduction – CADE-23*, 299–314. Berlin, Heidelberg: Springer Berlin Heidelberg.

Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2011. Sequential model-based optimization for general algorithm configuration. In *International conference on learning and intelligent optimization*, 507–523. Springer.

Jaskelioff, M., and Merz, S. 2005. Proving the correctness of disk paxos. *Archive of Formal Proofs*.

Kühlwein, D.; Blanchette, J. C.; Kaliszyk, C.; and Urban, J. 2013. Mash: Machine learning for sledgehammer. In Blazy, S.; Paulin-Mohring, C.; and Pichardie, D., eds., *Interactive Theorem Proving*, 35–50. Berlin, Heidelberg: Springer Berlin Heidelberg.

Matuszek, C.; Cabral, J.; and Wirbrock, M. 2006. An introduction to the syntax and content of Cyc. In Chitta, B., ed., *Formalizing and compiling background knowledge and its applications to knowledge representation and question answering. Papers from AAAI spring symposium*. Menlo Park, CA: AAAI Press. 44.

Močkus, J. 1975. On bayesian methods for seeking the extremum. In Marchuk, G. I., ed., *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, 400–404. Berlin, Heidelberg: Springer Berlin Heidelberg.

Nipkow, T.; Wenzel, M.; and Paulson, L. C. 2002. *Isabelle/HOL: A Proof Assistant for Higher-order Logic*. Berlin, Heidelberg: Springer-Verlag.

Rasmussen, C. E., and Williams, C. K. I. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Srinivas, N.; Krause, A.; Kakade, S.; and Seeger, M. 2010. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 1015–1022. USA: Omnipress.