Bayesian Optimisation with Gaussian Processes for Premise Selection

Agnieszka Słowik
University of Cambridge

Chaitanya Mangla
University of Cambridge

Mateja Jamnik
University of Cambridge

Sean B. Holden
University of Cambridge

Lawrence C. Paulson
University of Cambridge

Abstract

Heuristics in theorem provers are often parameterised. Modern theorem provers such as Vampire (Kovács & Voronkov, 2013) 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 probablistic framework for heuristics optimisation in theorem provers. We present results using a heuristic for premise selection and The Archive of Formal Proofs (AFP) (Jaskelioff & Merz, 2005) as a case study.

1 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. Such testing is often conducted on large theory sets, and is thus computationally expensive. Every assignment to the collection of parameters forms a point in parameter space, and as the parameters grow in number and range, an exhaustive search for optimal parameters becomes infeasible. An alternative is to sparsely navigate the space of parameters in search of the optimal point. We argue that probabilistic search enables efficient and automated optimisation of parameterised heuristics in theorem proving.

A simple probabilistic approach to parameter selection is to use a variation of ϵ-*greedy search* (Sutton & Barto, 2018). Given a metric that determines the value of each parameter combination within the parameter range, the agent starts at a random point and greedily searches the local neighbourhood. The best point found becomes the starting point for the next iteration. As a result, the agent is approaching a local optimum. However, with probability equal to ϵ, the agent randomly draws a new point for exploration. In theory, the ϵ-greedy search is an exhaustive search given unlimited time, and therefore in the limit it should give us the global optimum. In practice, it makes inefficient use of resources since the points it tests tend to be densely clustered. The key issue here is that the knowledge gained by testing any point is discarded.

In heuristics used in theorem proving, for instance premise selection algorithms, the objective function that maps parameter assignment to the number of proofs found is not given explicitly. What we care about is to find the parameter assignment that maximises the practical performance of a prover, and this should be done in a minimum number of evaluations to avoid the costs associated with exhaustive testing. The knowledge gained from every tested point in the parameter space can be used to reduce the uncertainty about the shape of the objective function. We can use our understanding of

the objective function within this uncertainty to make an informed decision about the next point to test. Every test further reduces our uncertainty. Furthermore, we often have a priori knowledge of certain aspects of the objective function. For example, we may have an idea of how close we expect neighbouring points to be. If we construct a probabilistic model of the objective function with this a priori knowledge, we can then use the knowledge gained from testing arbitrary points to increase our certainty, thereby improving our prediction of the next best point to test. *Bayesian Optimisation* (Močkus, 1975) is a principled method for this purpose and *Gaussian Processes* (Rasmussen & Williams, 2005) provide a means for probabilistic modelling of functions using prior knowledge and machine learning.

Here, we explore Bayesian Optimisation with Gaussian Processes 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 (Hoder & Voronkov, 2011). Our framework based on Gaussian Processes took up to 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 & Nipkow, 2010).

2   Premise Selection

In the experiments presented in this extended abstract we focused on the task of *premise selection* which can be defined as follows:

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

The size of the modern mathematical corpora creates a "needle in a haystack" problem for automated theorem provers. Only a small subset of available premises is relevant to any given conjecture. For instance, Open CYC (Matuszek et al., 2006) contains over 3 million axioms while each of the problems has a proof involving up to 20 premises.

Sumo Inference Engine (SInE) is a simple heuristic-based premise selection algorithm introduced to optimise reasoning in large theories. The algorithm aims to estimate the importance of function and predicate symbols based on their frequencies in the conjecture and in the premises at hand. The least frequent symbols indicate a *trigger relation* between the goal statement and a premise. This basic variant of the trigger relation is defined as follows:

Let $occ(s)$ be the number of premises in which the symbol s occurs and S the set of all symbols in a premise p. We define the least general symbol s' as the symbol for which:

$$\forall s \in S : occ(s') \leq occ(s) \tag{1}$$

and use it as a trigger for p. If the symbol s' appears in the conjecture, the algorithm selects the premise p. This basic heuristic suffers from low robustness since small changes in the number of frequencies can lead to a loss of important premises. The heuristic was thus extended in the following way:

$$trigger(s') \iff \{occ(s') \leq g\} \vee \{\forall s \in S : occ(s') \leq t \cdot occ(s)\} \tag{2}$$

where the parameters $t \geq 1$, $g \geq 1$ are referred to as the *tolerance threshold* and *generality threshold*, respectively.

Finally, premises triggered by the goal may contain symbols that lead to other relevant premises. This introduces another parameter $k \geq 0$ referred to as *depth*. It leads to the inductive construction of triggering symbols and premises:

1. All symbols s of the goal are 0-step triggered.

2. If s is k-step triggered and it triggers a premise p, then p is $k + 1$-step triggered.

3. If p is k-step triggered and s occurs in p, then s is k-step triggered.

The algorithm is therefore parameterised by one continuous parameter t and two discrete parameters g and k. As it was demonstrated in (Hoder & Voronkov, 2011), those parameters greatly influence the performance of the algorithm. The disparities between the optimal parameter assignment depending on the problem set can be significant. For instance, premise selection on SUMO and CYC problems from the TPTP library benefits from setting the depth parameter k to infinity, while using a problem set from Mizar results in selection of the sufficient number of premises with the depth limit set to 1. Bayesian optimisation framework provides an automatic suggestion of the optimal parameter assignment based on the problem set.

3   Bayesian Optimisation

Bayesian optimisation methods construct a probabilistic model of the objective function f and use it to determine informative sample locations. Under some prior on f, the points in the parameters space are repeatedly evaluated based on the posterior mean and variance predictions.

In our setting, the unknown objective function represents the usefulness of SInE given a parameter assignment. We assume this function was sampled from a Gaussian process. 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 Gaussian process which is one of the standard methods referred to as the *Gaussian Process-Upper Confidence Bound (GP-UCB)* algorithm (Srinivas et al., 2010). The subsections below describe this approach in more detail.

3.1   Problem statement

We want to maximise an unknown objective function $f : D \to \mathbb{R}$ which is correlated to the number of conjectures an ATP system would be able to prove given the premises selected by SInE. At each iteration i, we choose a point $x_i \in D$ and evaluate the current estimate of the objective function. The goal is to find the solution to the following expression (as rapidly as possible):

$$x^\star = \arg\max_{x \in D} f(x). \tag{3}$$

3.2   Gaussian Processes

A Gaussian Process is a distribution over functions specified by a mean μ and a covariance function κ:

$$f(x) \sim GP(\mu(x), \kappa(x, x')). \tag{4}$$

Common choices of covariance function include the finite dimensional linear, squared exponential and Matérn kernels (Duvenaud, 2014). We used the Matérn kernel which can be seen as a generalization of the Gaussian radial basis function.

3.3   Gaussian Process-Upper Confidence Bound

We define the *upper confidence bound* in the maximisation problem as:

$$UCB(x) = \mu(x) + \beta\sigma(x) \tag{5}$$

where $\beta \geq 0$. The first part of the update rule favours the points x which are likely to give a high reward in terms of the objective function. The second part prefers the points where the function f is uncertain, thereby negotiating the trade-off between exploitation and exploration. The amount of exploration is controlled by the constant β.

4 Experiments

4.1 Dataset

The Archive of Formal Proofs (AFP) (Jaskelioff & Merz, 2005) is a rapidly expanding[1] collection of proofs formalised in Isabelle (Nipkow et al., 2002). The repository is organized in the manner of a scientific journal. 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 (Böhme & Nipkow, 2010) that were used to determine which lemmas are needed to prove a goal.

4.2 Evaluation metrics

We have approximated the impact of premise selection on a theorem prover by using two evaluation metrics that are computed using SInE recommendations and the correct set of premises. The correct set of premises was obtained from Sledgehammer.

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 propose a metric based on *precision* and *recall*. In our setting, recall is represented by the ratio of the relevant premises recommended by the selection algorithm to the total number of premises needed for the proof. We experimented with several approaches to expressing precision in our setting and found that using a ratio of relevant premises recommended by SInE to an expression that increases exponentially with the size of the recommended set:

1. leads to an objective function that is strongly correlated with the main goal of recommending all of the premises needed to prove a conjecture;

2. favours a small number of redundant recommendations over the risk of missing one of the key premises, but penalises large sets of recommendations;

3. reduces sparsity that arises from directly optimising the number of theorems proved, which improves the optimisation process.

This metric is computed individually for each conjecture as follows:

Assuming we have a set of n conjectures, let $i = 1, 2, \ldots, n$ be the index of a conjecture to be proved. Let P_i be a non-empty set of lemmas required to prove the conjecture i and $\tilde{P}_i$ a set of lemmas recommended by a premise selection algorithm. Here, P_i is the set of premises used by Sledgehammer to prove the conjecture i, and $\tilde{P}_i$ refers to the set of premises recommended by SInE. If $|\tilde{P}_i| \neq 0$:

$$S_i = \frac{|\tilde{P}_i \cap P_i|}{|P_i|} + \frac{|\tilde{P}_i \cap P_i|}{2^{|\tilde{P}_i|}}. \tag{6}$$

For $|\tilde{P}_i| = 0$ we set the score S_i to zero since the conjecture could not be proved. After computing the value of S_i for each conjecture, we define the objective function across the whole dataset as follows:

$$S = \sum_{i=1}^{n} S_i. \tag{7}$$

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

---

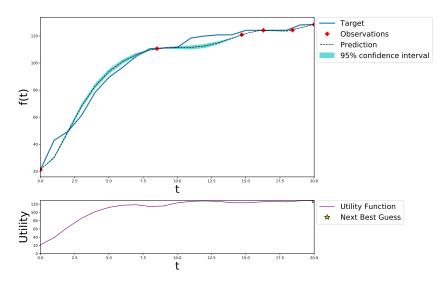[1]See the latest AFP statistics at `www.isa-afp.org/statistics.shtml`

Figure 1: Above: Approximation of the objective function (*target*) of one variable t and fixed discrete parameters ($g : 2, k : 5$), found after 3 random guesses and 3 iterations (*observations*). The function is increasing for $t \in (0, 20]$ so the optimisation problem in this range is trivial. Below: The corresponding value of the utility function (Upper Confidence Bound, see Subsection 3.3) and the current estimation of the global maximum.

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.

4.3 Results

4.3.1 Grid Search and Bayesian Optimisation

Gaussian Processes assume continuous input variables. Discrete-valued parameters require additional approximations. A common approach is to use a surrogate continuous variable for each discrete parameter, and round its value to the closest integer before evaluating the objective.

Instead of using approximations in the first set of experiments, we combined the exhaustive grid search approach for the two discrete parameters g, k with Bayesian optimisation of the continuous parameter t. In the objective function we observed a plateau for $t > 20$ and so we optimised the parameter in the $(0, 20]$ range (see Figure 1).

We compared this approach with ϵ-greedy search on the *Polynomials* article of 135 conjectures and with grid search on all of the three parameters (with t rounded to the nearest integer value). In this setting, the approach of employing Gaussian Processes gave the same results in terms of time and accuracy as ϵ-greedy search. This is due to the simplicity of the objective function given one variable t and the additional cost associated with the exhaustive search on parameters g and k. Both probabilistic approaches reached the global maximum faster than a grid search using all three parameters.

4.3.2 Bayesian Optimisation on the full set of parameters

In the next set of experiments, we used the standard approach of approximating the discrete parameters with continuous variables, and optimised the objective function on the full set of parameters (see Table 1).

The ϵ-search falls behind in this setting: we tested it for the three parameters within the time limits reported above, and the recommended parameters were in a close proximity to the initial random

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

Table 1: Premise selection results on the AFP articles. For each article we performed three iterations of the Bayesian optimisation process with two random starting points. We report the time needed to complete this fixed number of iterations – in several articles the optimum was found before the final iteration.

point, which yielded poor final results. We estimate that an exhaustive grid search in the comparable parameter space and on the same machine would take more than a day to find the global maximum.

5 Conclusion

The framework based on Bayesian optimisation with Guassian Processes turned out to be particularly effective in the multi-parameter setting. The main future goal is to evaluate the framework using Vampire and another mathematical corpus, for example TPTP (Sutcliffe, 2017).

References

Böhme, S. and Nipkow, T. Sledgehammer: Judgement day. In *Proceedings of the 5th International Conference on Automated Reasoning*, IJCAR'10, pp. 107–121, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-14202-8, 978-3-642-14202-4. doi: 10.1007/978-3-642-14203-1_9. URL `http://dx.doi.org/10.1007/978-3-642-14203-1_9`.

Duvenaud, D. *Automatic model construction with Gaussian processes*. PhD thesis, 11 2014.

Hoder, K. and Voronkov, A. Sine qua non for large theory reasoning. In Bjørner, N. and Sofronie-Stokkermans, V. (eds.), *Automated Deduction – CADE-23*, pp. 299–314, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-22438-6.

Jaskelioff, M. and Merz, S. Proving the correctness of disk paxos. *Archive of Formal Proofs*, June 2005. ISSN 2150-914x. `http://isa-afp.org/entries/DiskPaxos.html`, Formal proof development.

Kovács, L. and Voronkov, A. First-order theorem proving and vampire. In *Proceedings of the 25th International Conference on Computer Aided Verification - Volume 8044*, CAV 2013, pp. 1–35, New York, NY, USA, 2013. Springer-Verlag New York, Inc. ISBN 978-3-642-39798-1. doi: 10.1007/978-3-642-39799-8_1. URL `http://dx.doi.org/10.1007/978-3-642-39799-8_1`.

Kühlwein, D., Blanchette, J. C., Kaliszyk, C., and Urban, J. Mash: Machine learning for sledgehammer. In Blazy, S., Paulin-Mohring, C., and Pichardie, D. (eds.), *Interactive Theorem Proving*, pp. 35–50, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-39634-2.

Matuszek, C., Cabral, J., and Wirbrock, M. 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, pp. 44. AAAI Press, Menlo Park, CA, 2006.

Močkus, J. On bayesian methods for seeking the extremum. In Marchuk, G. I. (ed.), *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, pp. 400–404, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg. ISBN 978-3-540-37497-8.

Nipkow, T., Wenzel, M., and Paulson, L. C. *Isabelle/HOL: A Proof Assistant for Higher-order Logic*. Springer-Verlag, Berlin, Heidelberg, 2002. ISBN 3-540-43376-7.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. 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*, ICML'10, pp. 1015–1022, USA, 2010. Omnipress. ISBN 978-1-60558-907-7. URL http://dl.acm.org/citation.cfm?id=3104322.3104451.

Sutcliffe, G. The tptp problem library and associated infrastructure. *J. Autom. Reason.*, 59(4): 483–502, December 2017. ISSN 0168-7433. doi: 10.1007/s10817-017-9407-7. URL https://doi.org/10.1007/s10817-017-9407-7.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.