Estimating the Probability of a Conjecture to be a Theorem with PLN for Inference Control

Nil Geisweiller

SingularityNET Foundation,
Zug, Switzerland
nil@singularitynet.io

1 Introduction

The problem of estimating the probability of a conjecture to be a theorem has been mentioned in the literature as early as 1954 by George Pólya [6]. More formal treatments have appeared since then such as the work of Scott Garrabrant, Abram Demski et al on *Uniform Coherence* [1, 3] and *Logical Induction* [2], seemingly the most advanced treatments on the subject to date. To the best of our knowledge though, no one has considered using a probabilistic logic incorporating inductive and abductive reasoning such as NAL [7] or PLN [5], which we believe is particularly well suited to this problem. In this paper we show how to estimate the probability of a proposition to be a theorem given all available evidence by using Probabilistic Logic Networks (PLN). We then explain how such estimation can be used as guiding heuristics for Automated Theorem Proving.

The idea we develop here is to define a ternary predicate holding the relationship between theory, proof and theorem, and to probabilistically reason about it. Given its semi-decidable nature we cannot hope in practice to establish whether such relationship holds for any argument. We can however hope to estimate, with various degrees of confidence, the probability that it may or may not hold given the available evidence. For instance, some pieces of evidence in favor of

$$\forall x\ P(x)$$

could be that P holds for some $a_1, \dots, a_n$. Ideally every bit of information that relates to the conjecture should be taken into account to estimate its probability of being a theorem. Although only a rigorous proof, or a contradiction, can establish certainty. Moreover, such ability can then be used as a guide to discover proofs by prioritizing the search over lemmas that are themselves more likely to be provable. The same idea can be applied recursively on these lemmas till it bottoms out by reaching the axioms, or by exhibiting a contradiction, thereby hopefully reducing the amount of necessary backtracking.

2 Relating Theory, Proof and Proposition in PLN

From a type theoretic perspective, propositions are types, theories are collections of typing relationships and proofs are terms inhabiting types. Let us define a ternary predicate Θ, representing such relationship

$$\Theta : \texttt{Theory} \times \texttt{Proof} \times \texttt{Proposition} \rightarrow \texttt{Bool}$$

where `Theory` is a set of collections of typing relationships encoding the axioms and inference rules of each theory, `Proof` is a set of terms representing proofs, and `Proposition` is a set of types representing propositions. The content of Θ can in principle be characterized in PLN by formalizing the rewriting laws of such type system. For instance modus ponens could be formulated as[1]

$$\Theta(\Gamma, f, a \rightarrow b) \land \Theta(\Gamma, x, a) \Rightarrow \Theta(\Gamma, f(x), b) \ \stackrel{m}{=} \ <1,1>$$

where Γ, f, x, a and b are universally quantified variables, $\rightarrow$ is an implication at the object level, $\stackrel{m}{=}$ which can be read as *measured by* and relates a PLN *statement*, here an implication at the logical level $\Rightarrow$, to a *truth value*, here $<1,1>$, forming a PLN *judgment* capturing the uncertainty of the statement. The first number of the truth value represents the strength and the second number represents the confidence over that strength, although underneath, a truth value is a second order distribution. In the example above the judgment is certain because both the strength and the confidence are 1. The full judgment can be read as: in theory Γ, if f is a proof of $a \rightarrow b$, and x is a proof of a, then with certainty $f(x)$ is a proof of b. In addition, PLN allows to reason about uncertain knowledge via induction and abduction. Given a corpus of examples (and counter examples) of Θ, induction can be used to gather statistics about the probability of some propositions meeting some criteria to be theorems. Abduction provides a similar mechanism by considering properties over Θ instead of examples.

Given a theory Γ and a proposition C, the question *what is the probability that there exists a proof p of C in Γ?* can be formulated in PLN by the query

$$\exists p \ \Theta(\Gamma, p, C) \ \stackrel{m}{=} \ \$\mathtt{tv}$$

where $\$\mathtt{tv}$ is a hole corresponding to the truth value to be filled by PLN. The way this truth value is calculated may involve both crisp logical reasoning and statistical reasoning, the latter including recognizing patterns relating theories, proofs and propositions.

3 Estimating Provability as Guiding Heuristic

The ability to estimate the probability of finding a proof of a proposition could be used as guiding heuristic. For instance one may break up the task of finding a proof of C into two competing paths each composed of two subtasks:

- A-path: find a proof of $A \rightarrow C$, find a proof of A, then use modus ponens to prove C.
- B-path: find a proof of $B \rightarrow C$, find a proof of B, then use modus ponens to prove C.

To decide whether to take the A-path or the B-path, one may formulate the PLN queries

$$\exists p_A \ \Theta(\Gamma, p_A, A) \land \exists p_{AC} \ \Theta(\Gamma, p_{AC}, A \rightarrow C) \ \stackrel{m}{=} \ \$\mathtt{tv}_A$$

$$\exists p_B \ \Theta(\Gamma, p_B, B) \land \exists p_{BC} \ \Theta(\Gamma, p_{BC}, B \rightarrow C) \ \stackrel{m}{=} \ \$\mathtt{tv}_B$$

let PLN reason till both $\$\mathtt{tv}_A$ and $\$\mathtt{tv}_B$ get filled with truth values of decent confidences and pick up the path with the best truth value. If we want to compare the modus ponens rule to other inference rules, we can even go further by existentially quantifying the premise as well, resulting in the PLN query

$$\exists a \ \ (\exists p_a \ \Theta(\Gamma, p_a, a) \land \exists p_{aC} \ \Theta(\Gamma, p_{aC}, a \rightarrow C)) \ \stackrel{m}{=} \ \$\mathtt{tv}_a$$

If it gets selected the search will progressively instantiate a into actual premises, breaking up the query into more specific queries resembling the ones corresponding to the A-path and the B-path and so on. An early prototype of the ideas described in this paper can be found in [4].

[1] Note that due to PLN being in a state of rework, the syntax used here is provisional.

References

[1] Abram Demski. Logical Prior Probability. In Joscha Bach, Goerztel Ben, and Matthew Iklé, editors, *Artificial General Intelligence*, 2012.

[2] Scott Garrabrant, Tsvi Benson-Tilsen, Andrew Critch, Nate Soares, and Jessica Taylor. Logical Induction. *CoRR*, abs/1609.03543, 2016.

[3] Scott Garrabrant, Benya Fallenstein, Abram Demski, and Nate Soares. Uniform Coherence, 2016.

[4] Nil Geisweiller. PLN Inference Control Experiment in MeTTa. https://github.com/trueagi-io/chaining/tree/main/experimental/pln-inf-ctl.

[5] Ben Goertzel, Matthew Ikle, Izabela Freire Goertzel, and Ari Heljakka. *Probabilistic Logic Networks*. Springer US, 2009.

[6] George Pólya. *Mathematics and Plausible Reasoning Volume II: Patterns of Plausible Inference*. Princeton University Press, 1954.

[7] Pei Wang. *Non-Axiomatic Logic: A Model of Intelligent Reasoning*. World Scientific Publishing Co., Inc., USA, 2013.