Learning to Rank in Automatic Theorem Proving

Filip Bártek

Czech Technical University in Prague, Czech Republic
filip.bartek@cvut.cz

In saturation-based automatic theorem proving (ATP), clause selection is a crucial heuristic decision point. Neural networks (NNs) have been successfully trained to aid the proof search by approximately prioritizing clauses in a way that leads to a quick derivation of a proof [2, 6, 4, 8, 5]. Typically, such a NN is trained on a set of clauses that were derived in successful proof searches. The clauses are labeled: *Positive* clauses have contributed to a proof, while *negative* clauses have not. A straightforward approach trains a neural classifier of clauses [6, 8]. We motivate an alternative approach based on classification of clause *pairs* by two observations:

1. The training data in the form of labeled clauses can be interpreted as a specification of a *relative* preference over clauses: Each positive clause in a proof search is preferred over each negative clause in the same proof search. Specifying pairwise preference relation over clauses opens up the possibility of using finer-grained training data that only compares pairs of causes that belonged to a passive set simultaneously – we no longer have to consider every positive competing against every negative.

2. The final output of a clause selection heuristic is one clause from a passive set, rather than a partition of the passive set into positive and negative sets. Standard clause selection heuristics order the clauses by a heuristic weight (for example, derivation tree size or number of symbol occurrences in the clause) and prioritize clauses with the lowest weight. Similarly, we may prefer our NN to assign weights to clauses in such a way that the clauses with relatively small weights should be selected early.

Learning to rank is the machine learning task of training a ranking model – a system that ranks an arbitrary set of objects (e.g., clauses). The training data is typically supplied in the form of a partial order on a set of objects. *RankNet* [3] introduces a design of the loss function and the last layer of the NN that allows training the NN to rank arbitrary objects represented by feature vectors.

To train a clause selection heuristic in a RankNet-based approach, I trained a classifier of *clause pairs* C_+, C_- that estimates whether C_+ is more useful than C_- when these two clauses compete for selection in a proof search [2]. The NN predicts an intermediate weight $w(C)$ for each clause C. $w(C_+) < w(C_-)$ signifies that C_+ is estimated to be more useful than C_-, so ranking a set of clauses amounts to sorting the clauses by their predicted weights.

This design allowed me to define w as a weighted sum of symbol weights constrained to be greater than 1. The resulting clause weight was a pragmatic conservative modification of the popular symbol counting clause weight. Defining clause weight in this way would be non-trivial if the NN was trained as a clause classifier, while the RankNet design accommodates such definition naturally.

Notably, the RankNet-based approach is sufficiently generic to be applied to other decision points in ATP. In the past, I successfully applied it to symbol precedence recommendation [1].

In my presentation, I will explain the design of RankNet and its generalization DirectRanker [7], and describe how I applied it to clause selection and symbol precedence recommendation.

References

[1] Filip Bártek and Martin Suda. Neural precedence recommender. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings*, volume 12699 of *Lecture Notes in Computer Science*, pages 525–542. Springer, 2021.

[2] Filip Bártek and Martin Suda. How much should this symbol weigh? A gnn-advised clause selection. In Ruzica Piskac and Andrei Voronkov, editors, *LPAR 2023: Proceedings of 24th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Manizales, Colombia, 4-9th June 2023*, volume 94 of *EPiC Series in Computing*, pages 96–111. EasyChair, 2023.

[3] Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In Luc De Raedt and Stefan Wrobel, editors, *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pages 89–96. ACM, 2005.

[4] Karel Chvalovský, Jan Jakubuv, Martin Suda, and Josef Urban. ENIGMA-NG: efficient neural and gradient-boosted inference guidance for E. In Pascal Fontaine, editor, *Automated Deduction - CADE 27 - 27th International Conference on Automated Deduction, Natal, Brazil, August 27-30, 2019, Proceedings*, volume 11716 of *Lecture Notes in Computer Science*, pages 197–215. Springer, 2019.

[5] Karel Chvalovský, Konstantin Korovin, Jelle Piepenbrock, and Josef Urban. Guiding an instantiation prover with graph neural networks. In Ruzica Piskac and Andrei Voronkov, editors, *LPAR 2023: Proceedings of 24th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Manizales, Colombia, 4-9th June 2023*, volume 94 of *EPiC Series in Computing*, pages 112–123. EasyChair, 2023.

[6] Jan Jakubuv, Karel Chvalovský, Miroslav Olsák, Bartosz Piotrowski, Martin Suda, and Josef Urban. ENIGMA anonymous: Symbol-independent inference guiding machine (system description). In Nicolas Peltier and Viorica Sofronie-Stokkermans, editors, *Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part II*, volume 12167 of *Lecture Notes in Computer Science*, pages 448–463. Springer, 2020.

[7] Marius Köppel, Alexander Segner, Martin Wagener, Lukas Pensel, Andreas Karwath, and Stefan Kramer. Pairwise learning to rank by neural networks revisited: Reconstruction, theoretical analysis and practical performance. In Ulf Brefeld, Élisa Fromont, Andreas Hotho, Arno J. Knobbe, Marloes H. Maathuis, and Céline Robardet, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part III*, volume 11908 of *Lecture Notes in Computer Science*, pages 237–252. Springer, 2019.

[8] Martin Suda. Improving enigma-style clause selection while learning from history. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings*, volume 12699 of *Lecture Notes in Computer Science*, pages 543–561. Springer, 2021.