Project Proposal: Machine Learning Good Symbol Precedences[1]

Filip Bártek   Martin Suda

Czech Technical University in Prague, Czech Republic

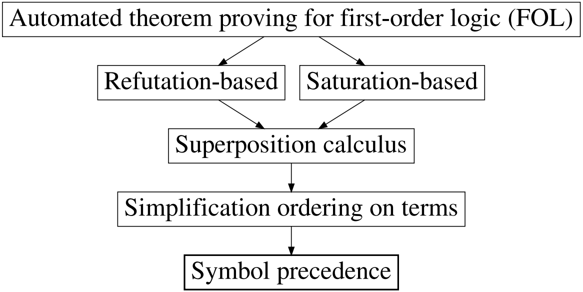
September 16, 2020

[1]Supported by the ERC Consolidator grant AI4REASON no. 649043 under the EU-H2020 programme, the Czech Science Foundation project 20-06390Y and the Grant Agency of the Czech Technical University in Prague, grant no. SGS20/215/OHK3/3T/37.

Outline

Context

Theorem prover of choice: Vampire



Why does symbol precedence matter?

FOL problem: $a = b \Rightarrow f(a, b) = f(b, b)$
CNF: $a = b \wedge f(a, b) \neq f(b, b)$

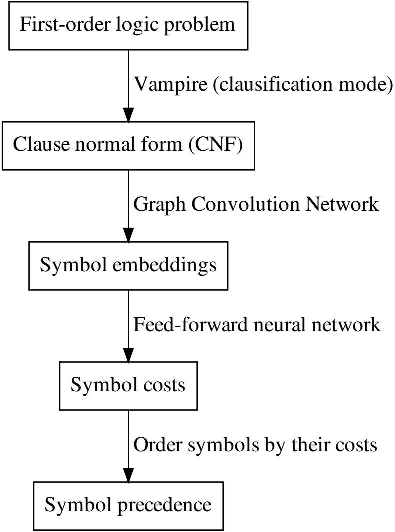
Precedence $[f, a, b]$ orders $a < b$:

$$
\begin{aligned}
&f(a, b) \neq f(b, b)\\
\to &f(a, a) \neq f(b, b)\\
\to &f(a, a) \neq f(a, b)\\
\to &f(a, a) \neq f(a, a)\\
\to &\bot
\end{aligned}
$$

Precedence $[f, b, a]$ orders $b < a$:

$$
\begin{aligned}
&f(a, b) \neq f(b, b)\\
\to &f(b, b) \neq f(b, b)\\
\to &\bot
\end{aligned}
$$

Precedence recommender system



Training data

Repeat:

1. Sample a problem P from TPTP.
2. Try to solve P using Vampire with two random precedences π_0, π_1.
3. If π_0 leads to a faster proof search than π_1, store the training sample (P, π_0, π_1).

We train a classifier that decides: Is π_0 better than π_1?

Model of "precedence π_0 is better than π_1"

1. Trainable symbol cost model $c_{sym} : \Sigma \to \mathbb{R}$
2. Precedence cost c_{prec} : $Precedences(\Sigma) \to \mathbb{R}$:

   $$c_{prec}(\pi) = \sum_{1 \leq i \leq |\Sigma|} c_{sym}(\pi(i)) \cdot i$$

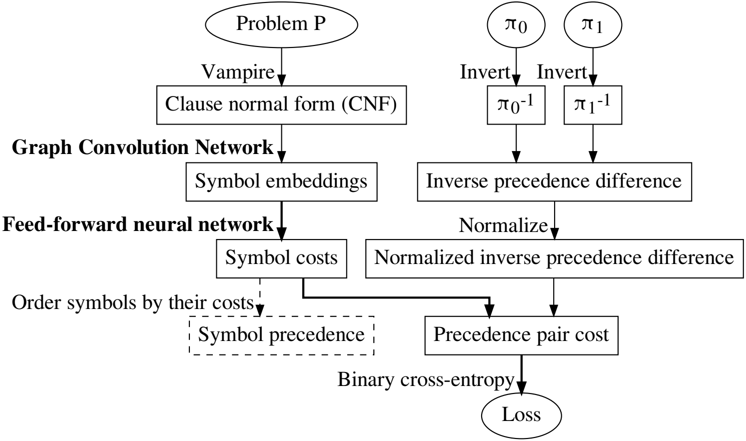
   Ordering symbols in decreasing order by c_{sym} minimizes c_{prec}.
3. Precedence pair cost:

   $$c_{pair}(\pi_0, \pi_1) = c_{prec}(\pi_1) - c_{prec}(\pi_0)$$

4. Probability that π_0 is better than π_1:

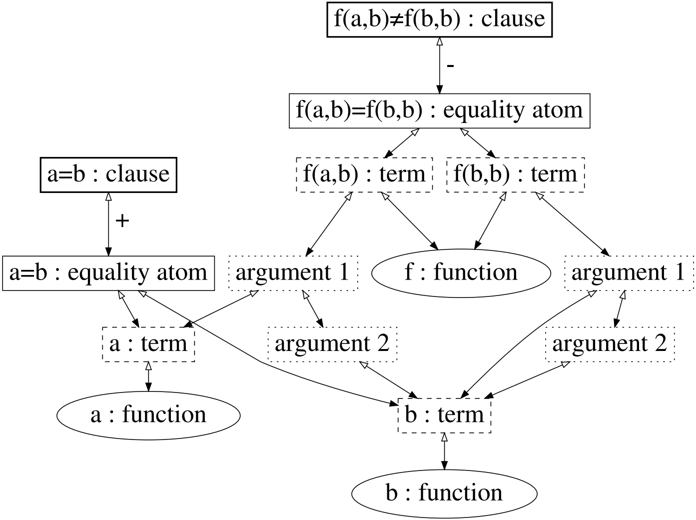
   $$\text{sigmoid}(c_{pair}(\pi_0, \pi_1))$$

Classifier: Is precedence π_0 better than π_1?



Graph Convolution Network example

$$a = b \wedge f(a, b) \neq f(b, b)$$



Preliminary experimental results

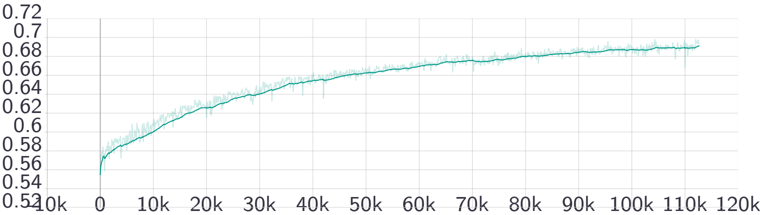


Figure: Accuracy versus training iterations

Symbol cost model	Accuracy
Graph Convolution Network	0.70
Frequency heuristic	0.56

Dataset: 4,821 problems, 1,411,730 precedence pairs

Section 4

Backup slides

Symbol costs rationale

Symbol cost function $c_{sym} : \Sigma \to \mathbb{R}$ is optimal on problem P iff ordering the symbols by their cost values in ascending order yields an optimal symbol precedence π^*.

This is true iff π^* minimizes $\sum_{1 \leq i \leq n} i \cdot c_{sym}(\pi(i))$ where $n = |\,\Sigma_P\,|$.

What is a good symbol cost function?

How can we train symbol costs such that when we order symbols by symbol costs

Training data

Model layers:

1. Problem -> symbol embeddings 2. Symbol embedding -> symbol cost 3. Symbol costs -> precedence cost

Let $s \in \Sigma$.

Let M_c be a differentiable symbol cost model: $c_{sym}(s) = M_c(fv(s))$

$$c_{prec}(\pi) = C \sum_{1 \leq i \leq n} c_{sym}(\pi(i)) \cdot i = C \sum_{1 \leq i \leq n} c_{sym}(s_i) \cdot \pi^{-1}(s_i)$$

$$c_{prec}(\pi) = C \sum_{1 \leq i \leq n} c_{sym}(\pi(i)) \cdot f(i) = C \sum_{1 \leq i \leq n} c_{sym}(s_i) \cdot f(\pi^{-1}(s_i))$$

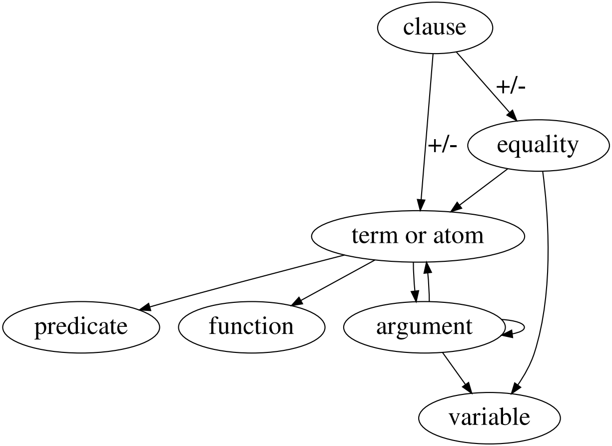
$C = \frac{2}{n(n+1)}$ so that $c_{sym}(s) = 1$ for all s implies $c_{prec}(\pi) = 1$ for all π.

$$c_{pair}(\pi_0, \pi_1) = c_{prec}(\pi_1) - c_{prec}(\pi_0) = C \sum_{1 \leq i \leq n} c_{sym}(s_i) \cdot [\pi_1^{-1}(s_i) - \pi_0^{-1}(s_i)]$$

Loss: L()...

Our math model of precedence cost: weighted sum of symbol costs. Show on an example that minimizing this expression corresponds to sorting in descending order.
We search for csym such that cprec correlates with the quality of precedence.
Why pairs of precedences? We are sure which of two is better but we are not sure what is a good (target) quality value of a precedence.

Graph Convolution Network schema



Symbol features: in conjecture, introduced

GNN architecture

Trainable parameters are *emphasized*.

- For each node type: *layer 0 node embedding*
- For each layer:
  - For each edge type: *Message model* (dense layer)
    - Input: source node embedding, source node features, edge features
    - Output: message
  - Message aggregation step (sum all incoming messages for each node and incoming edge type)
  - For each node type: *Node aggregation model* (dense layer)
    - Input: node embedding, aggregated message for each incoming edge type
    - Output: node embedding

References

Geoff Sutcliffe. The TPTP problem library and associated infrastructure. From CNF to TH0, TPTP v6.4.0. *Journal of Automated Reasoning*, 59(4):483–502, 2017. doi: 10.1007/s10817-017-9407-7.

Experimental setup

- Only predicate precedences are learned.
  Function symbols are ordered by `invfreq`.
- Problems from TPTP Sutcliffe [2017] – CNF and FOF (clausified with Vampire)
  - $\mathcal{P}_{train}$ (8217 problems): at most 200 predicate symbols, at least 1 out of 24 random predicate precedences yield success
  - $\mathcal{P}_{test}$ (15751 problems): at most 1024 predicate symbols
- 5 evaluation iterations (splits): 1000 training problems and 1000 test problems
- 100 precedences per training problem
- Vampire configuration: time limit: 10 seconds, memory limit: 8192 MB, literal comparison mode: predicate, function symbol precedence: `invfreq`, saturation algorithm: discount, age-weight ratio: 1:10, AVATAR: disabled
- 10^6 symbol pair samples to train M

Elastic-Net feature coefficients

of individual symbols

Training set	Arity	Frequency	Unit frequency
0	$-.98$	.01	$-.01$
1		.56	.44
2		.36	.64
3	$-.88$		.04
4		.93	.07
$\mathcal{P}_{train}$		.43	.57

Symbol order: descending by predicted value

- Sets 1, 2, 4, $\mathcal{P}_{train}$:
  - Descending by frequency: low frequency $\sim$ early inference
  - Similar to `invfreq` and `vampire --sp frequency`
- Sets 0, 3:
  - Ascending by arity: high arity $\sim$ early inference
  - Similar to `vampire --sp arity`