Learning-Assisted Automated Reasoning

Lecture 3

Cezary Kaliszyk

October 20, 2018

Summary

Last time

Theorem proving systems

Machine learning problems

Lemma relevance

Features and deep approaches

(ordered) Resolution, Superposition

Today

Learning in E-Prover

Learning in Tableaux

Reinforcement Learning

Machine learning in proof techniques already

Relevant Knowledge Selection

- Automated Reasoning
- Important component for human-computer interaction

Conjecturing / Theory Exploration

- Statistical and Generative approaches

...

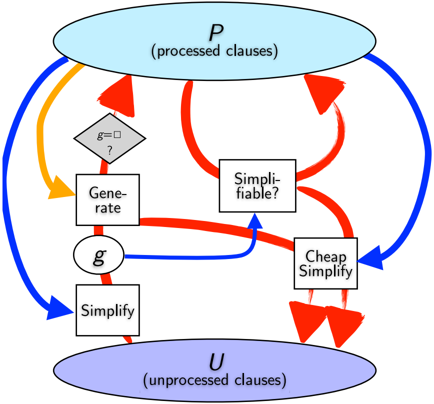
More human-like proof (divide and conquer) (?)

Auto-formalization (?)

Modern superposition-based prover

- The calculus
- Subsumption
- Good indexing
- Tautology detection (also for equality)
- Heuristics for orderings, selection
- Implementation (engineering)
- ...

E-Prover given-clause loop



Most important choice: unprocessed clause selection

[Schulz'15]

Data Collection

Mizar top-level theorems

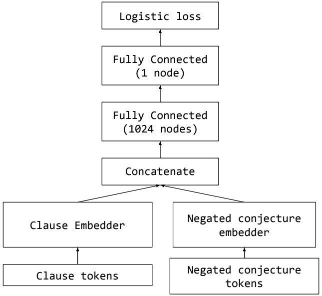
- Encoded in FOF

32,521 Mizar theorems with ≥ 1 proof

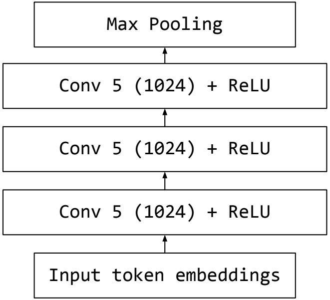
- training-validation split (90%-10%)
- replay with one strategy

Collect all CNF intermediate steps

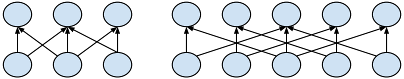
Deep Network Architectures



Overall network



Convolutional Embedding



Dilatated convolutions

Recursive Neural Networks

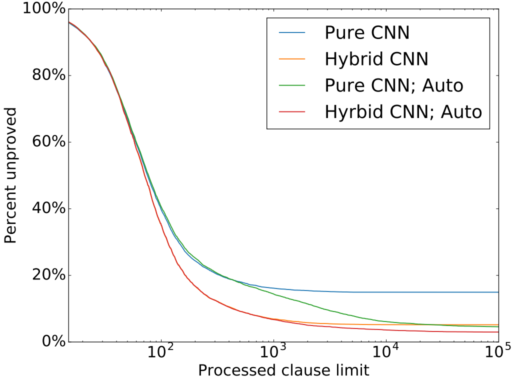
- Curried representation of first-order statements
- Separate nodes for `apply`, `or`, `and`, `not`
- Layer weights learned jointly for the same formula
- Embeddings of symbols learned with rest of network
- Tree-RNN and Tree-LSTM models

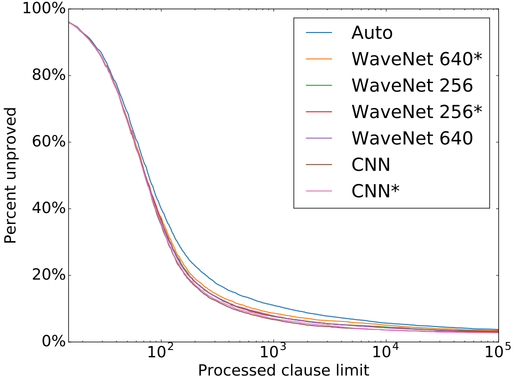
Model accuracy

Model	Embedding Size	Accuracy on 50-50% split
Tree-RNN-256×2	256	77.5%
Tree-RNN-512×1	256	78.1%
Tree-LSTM-256×2	256	77.0%
Tree-LSTM-256×3	256	77.0%
Tree-LSTM-512×2	256	77.9%
CNN-1024×3	256	80.3%
*CNN-1024×3	256	78.7%
CNN-1024×3	512	79.7%
CNN-1024×3	1024	79.8%
WaveNet-256×3×7	256	79.9%
*WaveNet-256×3×7	256	79.9%
WaveNet-1024×3×7	1024	81.0%
WaveNet-640×3×7(20%)	640	**81.5%**
*WaveNet-640×3×7(20%)	640	79.9%

Hybrid Heuristic

Already on proved statements performance requires modifications:





Harder Mizar top-level statements

Model	DeepMath 1	DeepMath 2	Union of 1 and 2
Auto	578	581	674
*WaveNet 640	644	612	767
*WaveNet 256	692	712	864
WaveNet 640	629	685	997
*CNN	905	812	1,057
CNN	839	935	1,101
Total (unique)	1,451	1,458	1,712

Overall proved 7.36% of the harder statements

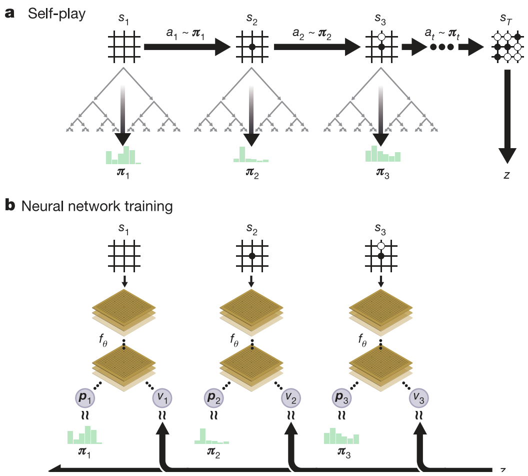
Learn reasoning step selection

What does AlphaZero do? [*Silver et al.*]

- Self-play (learned strategy)
- Repeated strategy improvement
- Search game tree wisely
- Estimate moves and states

Can we do this for theorem proving?

What went wrong with E-prover?



leanCoP: Lean Connection Prover

[Otten 2010]

Connected tableaux calculus

- Goal oriented, good for large theories

Regularly beats Metis and Prover9 in CASC (CADE ATP competition)

- despite their much larger implementation

Compact Prolog implementation, easy to modify

- Variants for other foundations: iLeanCoP, mLeanCoP
- First experiments with machine learning: MaLeCoP

Easy to imitate

- leanCoP tactic in HOL Light

Lean connection Tableaux

Very simple rules:

- Extension unifies the current literal with a copy of a clause
- Reduction unifies the current literal with a literal on the path

$$\frac{}{\{\},\ M,\ Path} \quad \textit{Axiom}$$

$$\frac{C,\ M,\ Path \cup \{L_2\}}{C \cup \{L_1\},\ M,\ Path \cup \{L_2\}} \quad \textit{Reduction}$$

$$\frac{C_2 \setminus \{L_2\},\ M,\ Path \cup \{L_1\} \qquad C,\ M,\ Path}{C \cup \{L_1\},\ M,\ Path} \quad \textit{Extension}$$

Example lean connection proof

Clauses:

$c_1 : P(x)$

$c_2 : R(x, y) \vee \neg P(x) \vee Q(y)$

$c_3 : S(x) \vee \neg Q(b)$

$c_4 : \neg S(x) \vee \neg Q(x)$

$c_5 : \neg Q(x) \vee \neg R(a, x)$

$c_6 : \neg R(a, x) \vee Q(x)$

Tableau:

- $P(a)$
  - $R(a, b)$
    - $\neg R(a, b)$
    - $Q(b)$
      - $\neg Q(b)$
      - $\neg R(a, b)$
  - $\neg P(a)$
  - $Q(b)$
    - $S(b)$
      - $\neg S(b)$
      - $\neg Q(b)$
    - $\neg Q(b)$

First experiment: MaLeCoP in Prolog

[*Tableaux 2011*]

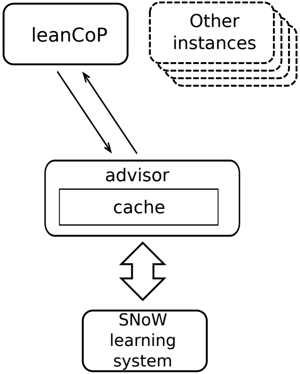
Select extension steps

- Using external advice

Slow implementation

- 1000 times less inf per second

Can avoid 90% inferences!



What about efficiency: FEMaLeCoP [LPAR 2015]

Very simple but very fast classifier built-in

- Naive Bayes (with optimizations)

Approximate features and multi-level indexing

- Offline indexing
- Indexing for the current problem
- Discrimination tree stores NB data

Consistent clausification and skolemization

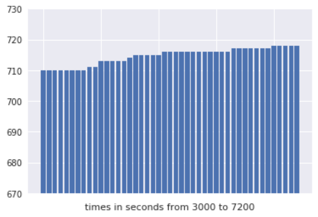
Performance is about 40% of non-learning leanCoP speed

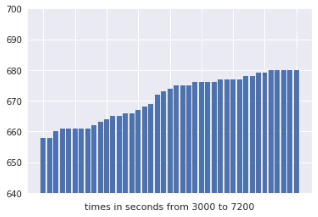
- A few more theorems proved (3–15%)

What about stronger learning?

Yes, but...

- If put directly, huge times needed
- Still improvement small



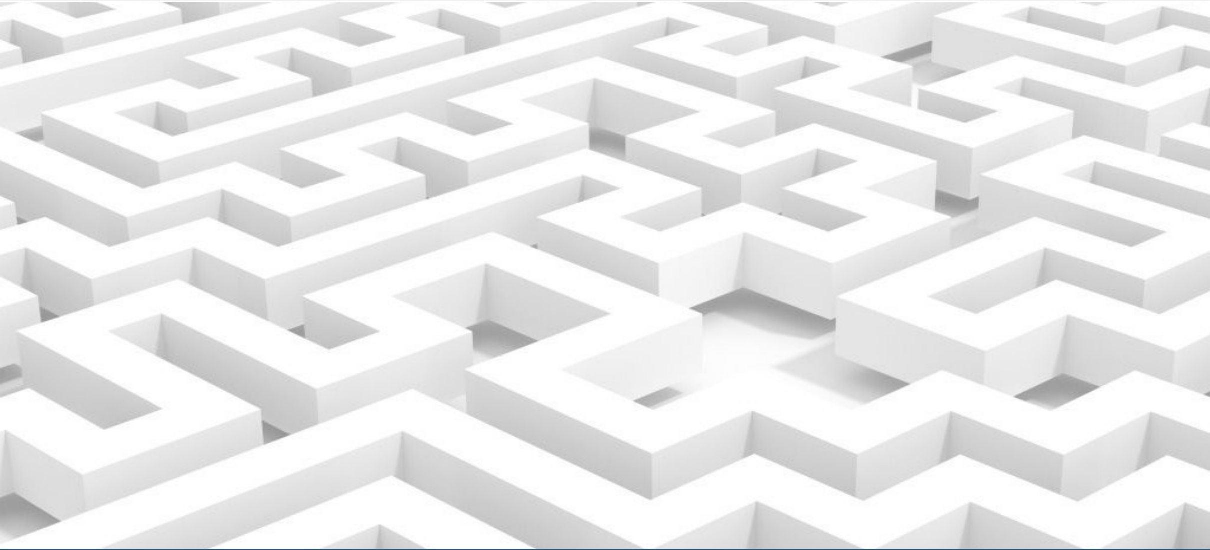


NBayes vs XGBoost on 2h timeout

Preliminary experiments with deep learning

- So far quite slow

Could we give our tableaux to an AI / game engine?



Is theorem proving just a maze search?

Yes and NO!

- The proof search tree is not the same as the tableau tree!
- Unification can cause other branches to disappear.

Provide an external interface to proof search

- Usable in OCaml, C++, and Python
- Two functions suffice

$$\text{start} : \text{problem} \to \text{state}$$

$$\text{go} : \text{action} \to \text{state}$$

- where

$$\text{state} = \langle \text{avail action list} \times \text{remaining goal-paths} \rangle$$

Is it ok to change the tree?

Most learning for games sticks to game dynamics

- Only tell it how to do the moves

Why is it ok to skip other branches

- Theoretically ATP calculi are complete
- Practically most ATP strategies incomplete

In usual 30s – 300s runs

- Depth of proofs with backtracking: 5–7 (complete)
- Depth with restricted backtracking: 7–10 (more proofs found!)

But with random playouts: depth hundreds of thousands!

- Just unlikely to find a proof $\rightarrow$ learning

Monte Carlo First Try: monteCoP

[CADE'17]

Use Monte Carlo playouts to guide restricted backtracking

- Improves on leanCoP, but not by a big margin
- Potential still limited by depth

What could we do more?

- Learn both policy and value
- Unfold only useful branches
- Do not backtrack
- Arbitrarily long playouts
- Learn both proofs and lack thereof

How to search a tree? [Szepesvari 2006]

Monte Carlo Tree Search

Upper Confidence Bounds for Trees

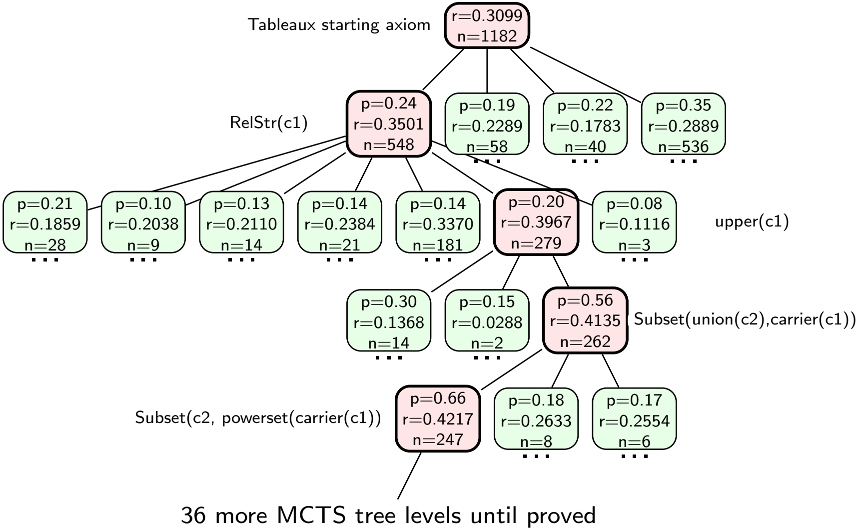
UCT: Select node n maximizing

$$\frac{w_i}{n_i} + c \cdot p_i \cdot \sqrt{\frac{\ln N}{n_i}}$$

Intuition

- Initially proportional to the prior
- Later average reward dominates
- Heuristic replaced by learned priors and rewards

MCTS tree for WAYBEL_0:28



Learn Policy and Value

Policy: Which actions to take?

- Proportions predicted based on proportions in similar states
- Explore less the actions that were "bad" in the past
- Explore more and earlier the actions that were "good"

Value: How good (close to a proof) is a state?

- Reward states that have few goals
- Reward easy goals

Where to get training data?

- Explore 1000 nodes using UCT
- Select the most visited action and focus on it for this proof
- A sequence of selected actions can train both policy and value

Initial comparison: 2000 selected easier Mizar Problems

Baseline, in 200K inferences

leanCoP	random playouts	plain UCT
876	434	770

Guided by reinforcement learning from previous iterations (UCT)

Iteration	1	2	3	4	5	6	7	8	9	10
Proved	1037	1110	1166	1179	1182	1198	1196	1193	1212	1210
Iteration	11	12	13	14	15	16	17	18	19	20
Proved	1206	1217	1204	1219	1223	1225	1224	1217	1226	**1235**

Proper Evaluation: Train (29272) and test (3252) sets

Baseline

System	leanCoP	playouts	UCT
Train	10438	4184	7348
Test	**1143**	431	804

10 iterations

Iteration	1	2	3	4	5	6	7	8	9
Train	12325	13749	14155	14363	14403	14431	14342	**14498**	14481
Test	1354	1519	1566	1595	**1624**	1586	1582	1591	1577

More Time

leanCoP, 4M inferences, strategies	1396
rlCoP union	1839