Machine Learning Problems in Automated Theorem Proving

Cezary Kaliszyk

CL Seminar

October 18, 2017

Mathematicians and Proof Assistants

Plenty of Improvements in the last decade

- But same complaint: Mathematicians do not use PAs

Finally huge success stories

- But we can count them on our fingers

Issues

Lack of Libraries

- what if we just assume properties?

Automation

- with incomplete proofs?

Many Foundations and Systems

- Wiedijk's "QED Revised" (2007)
  - *"none of the [. . .] systems can express all four statements in a good way"*
- But many domains work in any foundation

Communication

- The prover does not get what I mean

Fast progress in machine learning

Tasks involving logical inference

- Natural language question answering [*Sukhbaatar+2015*]
- Knowledge base completion [*Socher+2013*]
- Automated translation [*Wu+2016*]

Games

AlphaGo problems similar to proving [*Silver+2016*]

- Node evaluation
- Policy decisions

Outline

TP ML Problems

State Evaluation

Learning in ATPs

AI theorem proving techniques

High-level AI guidance

- premise selection: select the right lemmas to prove a new fact
- based on suitable features (characterizations) of the formulas
- and on learning lemma-relevance from many related proofs
- tactic selection

Mid-level AI guidance

- learn good ATP strategies/tactics/heuristics for classes of problems
- learning lemma and concept re-use
- learn conjecturing

Low-level AI guidance

- guide (almost) every inference step by previous knowledge
- good proof-state characterization and fast relevance

Problems for Machine Learning

- Is a statement is useful?
  - For a conjecture
- What are the dependencies of statement? (premise selection)
- Is a statement important? (named)
- What should the next proof step be?
  - Tactic? Instantiation?
- How to name a statement?
- What new problem is likely to be true?
  - Intermediate statement for a conjecture

Premise Selection

- Syntactic methods
  - Neighbours using various metrics
  - Recursive: SInE, MePo
- Naive Bayes, k-Nearest Neighbours
- Regression
  - Needs feature and theorem space reduction
  - Kernel-based multi-output ranking
- Decision Trees (Random Forests)
- Neural Networks
  - Winnow, Perceptron (SNoW)
  - DeepMath

Outline

TP ML Problems

State Evaluation

Learning in ATPs

Intermediate lemmas

Size of inference graphs

	HOL Light graph		Flyspeck graph	
	nodes	edges	nodes	edges
kernel inferences	8,919,976	10,331,922	1,728,861,441	1,953,406,411
reduced trace	2,076,682	3,002,990	159,102,636	233,488,673
tactical inferences	148,514	594,056	11,824,052	42,296,208
tactical trace	22,284	89,981	1,067,107	4,268,428

Why not use these lemmas?

- The graphs are already computed outsize of HOL
- Feature extraction, prediction, translation do not scale...
- Pre-select heuristically interesting lemmas

Metric based lemmas

Definition (Recursive dependencies and uses)

$$D(i) = \begin{cases} 1 & \text{if } i \in Named \vee i \in Axioms, \\ \sum_{j \in d(i)} D(j) & \text{otherwise.} \end{cases}$$

Definition (Lemma quality)

$$Q_1(i) = \frac{U(i) * D(i)}{S(i)} \qquad Q_1^r(i) = \frac{U(i)^r * D(i)^{2-r}}{S(i)}$$

$$Q_2(i) = \frac{U(i) * D(i)}{S(i)^2} \qquad Q_3(i) = \frac{U(i) * D(i)}{1.1^{S(i)}}$$

EpclLemma (longest chain), AGIntRater, ...

PageRank and Cut

- Pagerank: Fast, non-iterative, usable on whole Flyspeck
- Dominant eigenvector of:

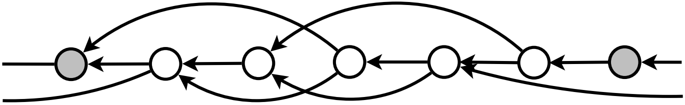
$$PR_1(i) = \frac{1-f}{N} + f \sum_{i \in d(j)} \frac{PR_1(j)}{|d(j)|}$$

- Size normalized

$$PR_2(i) = \frac{PR_1(i)}{S(i)}$$

Maximum Graph Cut

Gray nodes correspond to named theorems:



PageRank and Cut

- Pagerank: Fast, non-iterative, usable on whole Flyspeck
- Dominant eigenvector of:

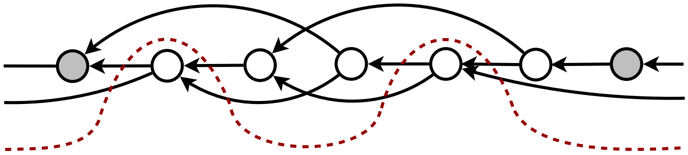
$$PR_1(i) = \frac{1-f}{N} + f \sum_{i \in d(j)} \frac{PR_1(j)}{|d(j)|}$$

- Size normalized

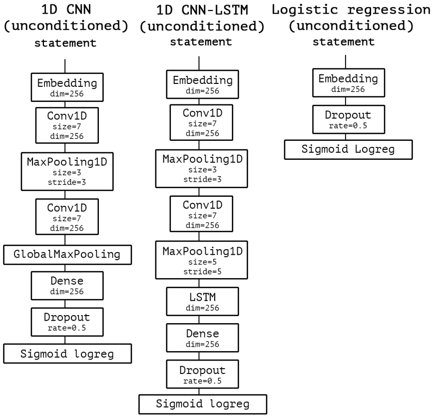
$$PR_2(i) = \frac{PR_1(i)}{S(i)}$$

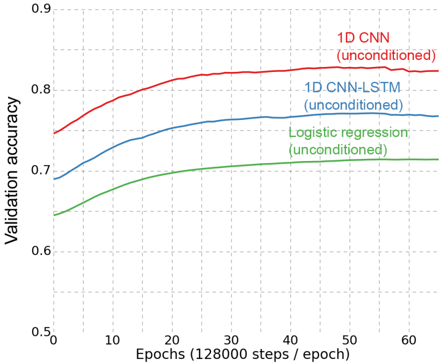
Maximum Graph Cut

Gray nodes correspond to named theorems:



Deep Learning





Outline

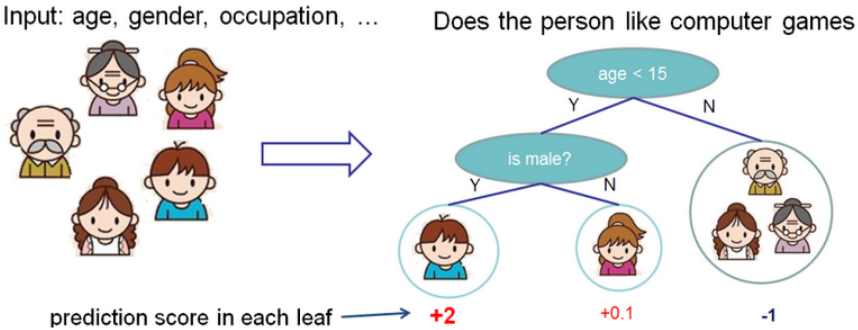
leanCoP: Lean Connection Prover (Jens Otten)

- Connected tableaux calculus
  - Goal oriented, good for large theories
- Regularly beats Metis and Prover9 in CASC
  - despite their much larger implementation
  - very good performance on some ITP challenges
- 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

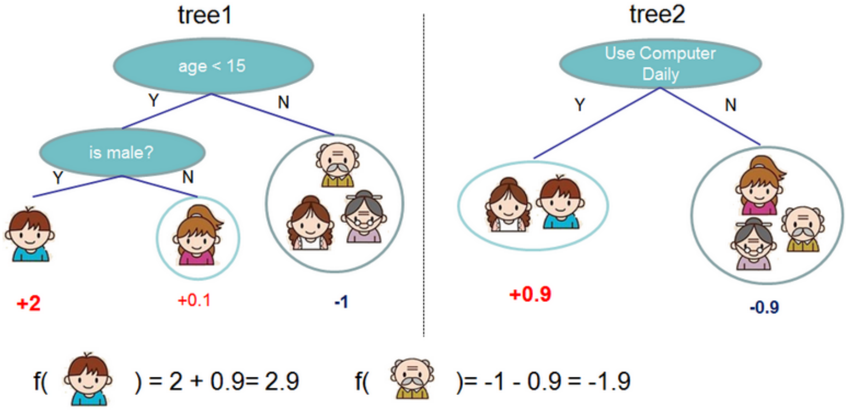
FEMaLeCoP: Advice Overview and Used Features

- Advise the:
  - **selection of clause for every tableau extension step**
- Proof state: weighted vector of symbols (or terms)
  - extracted from all the literals on the active path
  - Frequency-based weighting (IDF)
  - Simple decay factor (using maximum)
- Consistent clausification
  - formula `?[X]: p(X)` becomes `p('skolem(?[A]:p(A),1)')`
- Advice using custom sparse naive Bayes
  - association of the features of the proof states
  - with contrapositives used for the successful extension steps
- Data Collection and Indexing

XGBoost Forests (1/2)



XGBoost Forests (2/2)



Number of proved theorems

- OCaml leanCoP: 680 theorems
- leanCoP with Naive Bayes guidance: 720 theorems (can be improved)
- leanCoP with XGBoost guidance trained on the top of 720 theorems: 777 theorems with some ensembling (can be improved)

Ongoing:

- More Monte Carlo *[MF+CADE'17]*
- Simple Reinforcement *[Dagger]*
- Full Reinforcement Learning
    - Lots of issues

Summary

- Many ITP/ATP problems could be interesting for AI
- Stronger techniques often too slow to be practical for ITPs
  - Space reductions and approximations make algorithms weaker
- Next step: Internal guidance for Automated Theorem Proving
  - Fast learning algorithm, indexing, approximate features
- Finally: Generate interesting conjectures and proofs
  - "Replace mathematicians"