Premise Selection, Hammers, Features

Josef Urban

Czech Technical University in Prague

May 10, 2019

Mizar demo

http://grid01.ciirc.cvut.cz/~mptp/out4.ogv

Using Learning to Guide Theorem Proving

- **high-level**: pre-select lemmas from a large library, give them to ATPs
- **high-level**: pre-select a good ATP strategy/portfolio for a problem
- **high-level**: pre-select good *hints* for a problem, use them to guide ATPs
- **low-level**: guide every inference step of ATPs (tableau, superposition)
- **low-level**: guide every kernel step of LCF-style ITPs
- **mid-level**: guide application of tactics in ITPs
- **mid-level**: invent suitable ATP strategies for classes of problems
- **mid-level**: invent suitable conjectures for a problem
- **mid-level**: invent suitable concepts/models for problems/theories
- **proof sketches**: explore stronger/related theories to get proof ideas
- **theory exploration**: develop interesting theories by conjecturing/proving
- **feedback loops**: (dis)prove, learn from it, (dis)prove more, learn more, ...
- ...

Sample of Learning Approaches We Have Been Using

- **neural networks** (statistical ML) – backpropagation, deep learning, convolutional, recurrent, etc.
- **decision trees, random forests, gradient tree boosting** – find good classifying attributes (and/or their values); more explainable
- **support vector machines** – find a good classifying hyperplane, possibly after non-linear transformation of the data (*kernel methods*)
- **k-nearest neighbor** – find the k nearest neighbors to the query, combine their solutions
- **naive Bayes** – compute probabilities of outcomes assuming complete (naive) independence of characterizing features (just multiplying probabilities)
- **inductive logic programming** (symbolic ML) – generate logical explanation (program) from a set of ground clauses by generalization
- **genetic algorithms** – evolve large population by crossover and mutation
- combinations of statistical and symbolic approaches (probabilistic grammars, semantic features, ...)
- supervised, unsupervised, reinforcement learning (actions, explore/exploit, cumulative reward)

Learning – Features and Data Preprocessing

- Extremely important - if irrelevant, there is no use to learn the function from input to output (“garbage in garbage out”)
- Feature discovery – a big field
- Deep Learning – design neural architectures that automatically find important high-level features for a task
- Latent Semantics, dimensionality reduction: use linear algebra (eigenvector decomposition) to discover the most similar features, make approximate equivalence classes from them
- word2vec and related methods: represent words/sentences by *embeddings* (in a high-dimensional real vector space) learned by predicting the next word on a large corpus like Wikipedia
- math and theorem proving: syntactic/semantic patterns/abstractions
- how do we represent math objects (formulas, proofs, ideas) in our mind?

Reasoning Datasets - Large ITP Libraries and Projects

- Mizar / MML / MPTP – since 2003
- MPTP Challenge (2006), MPTP2078 (2011), Mizar40 (2013)
- Isabelle (and AFP) – since 2005
- Flyspeck (including core HOL Light and Multivariate) – since 2012
- HOLStep – 2016, kernel inferences
- Coq – since 2013/2016
- HOL4 – since 2014
- ACL2 – 2014?
- Lean? – 2017?
- Stacks?, ProofWiki?, Arxiv?

High-level ATP guidance: Premise Selection

- Early 2003: Can existing ATPs be used over the freshly translated Mizar library?
- About 80000 nontrivial math facts at that time – impossible to use them all
- Is good premise selection for proving a new conjecture possible at all?
- Or is it a mysterious power of mathematicians? (Penrose)
- Today: Premise selection is not a mysterious property of mathematicians!
- Reasonably good algorithms started to appear (more below).
- Will extensive human (math) knowledge get obsolete?? (cf. Watson, Debater, etc)

Example system: Mizar Proof Advisor (2003)

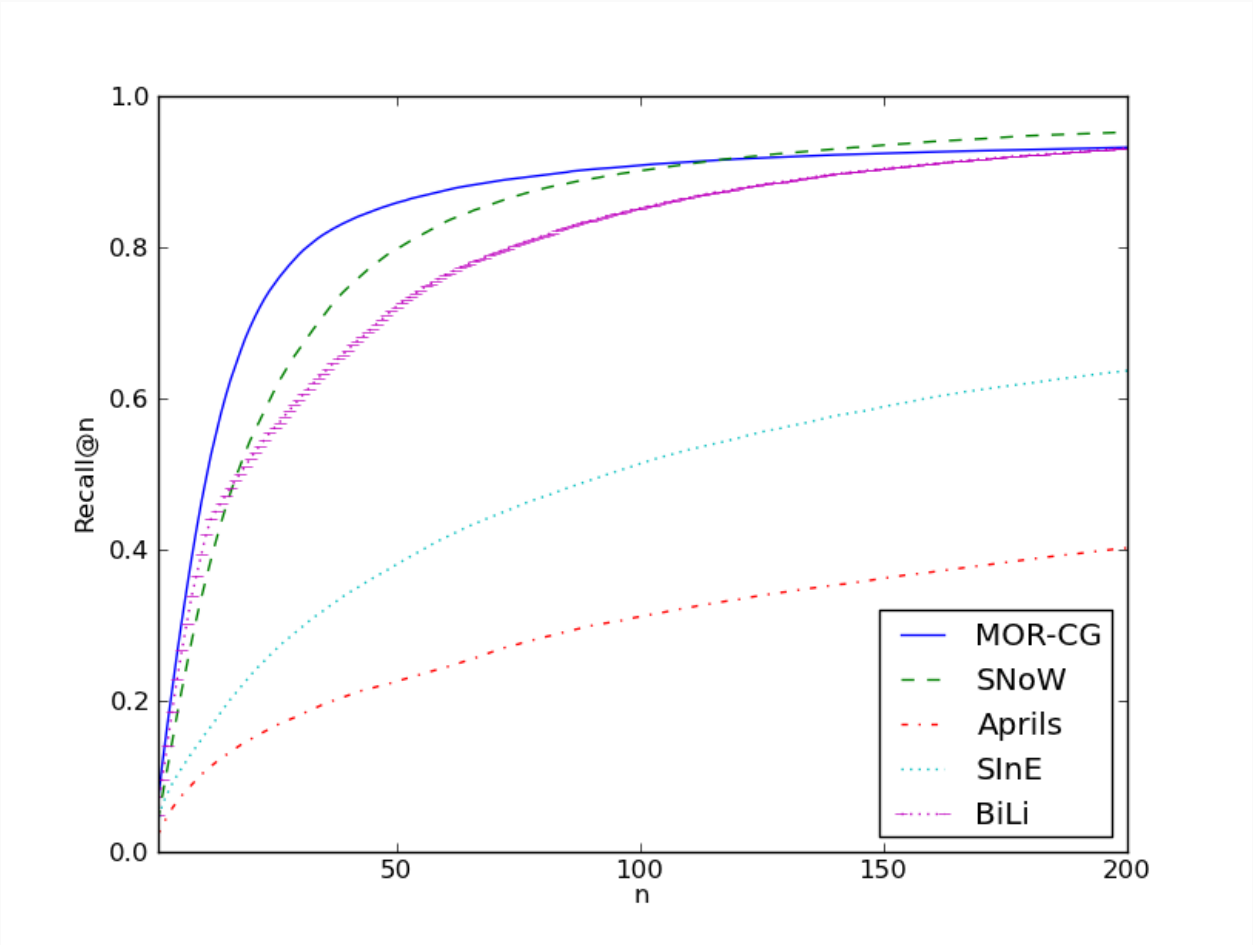
- train naive-Bayes fact selection on all previous Mizar/MML proofs (50k)
- input features: conjecture symbols; output labels: names of facts
- recommend relevant facts when proving new conjectures
- give them to unmodified FOL ATPs
- possibly reconstruct inside the ITP afterwards (lots of work)
- First results over the whole Mizar library in 2003:
  - about 70% coverage in the first 100 recommended premises
  - chain the recommendations with strong ATPs to get full proofs
  - about 14% of the Mizar theorems were then automatically provable (SPASS)
- Today's methods: about 45-50% (and we are still just beginning!)

Smaller AI/ATP benchmarks: MPTP Challenge (2006)

- 252 problems from Mizar – Bolzano-Weierstrass theorem
- small (bushy) and large (chainy) problems
- about 1500 formulas altogether
- a bigger version in 2011: 2078 problems, 4500 formulas – MPTP2078
- large-theory reasoning competitions: CASC LTB (since 2008)
- Large Mizar benchmark: Mizar40 – about 60k Mizar problems

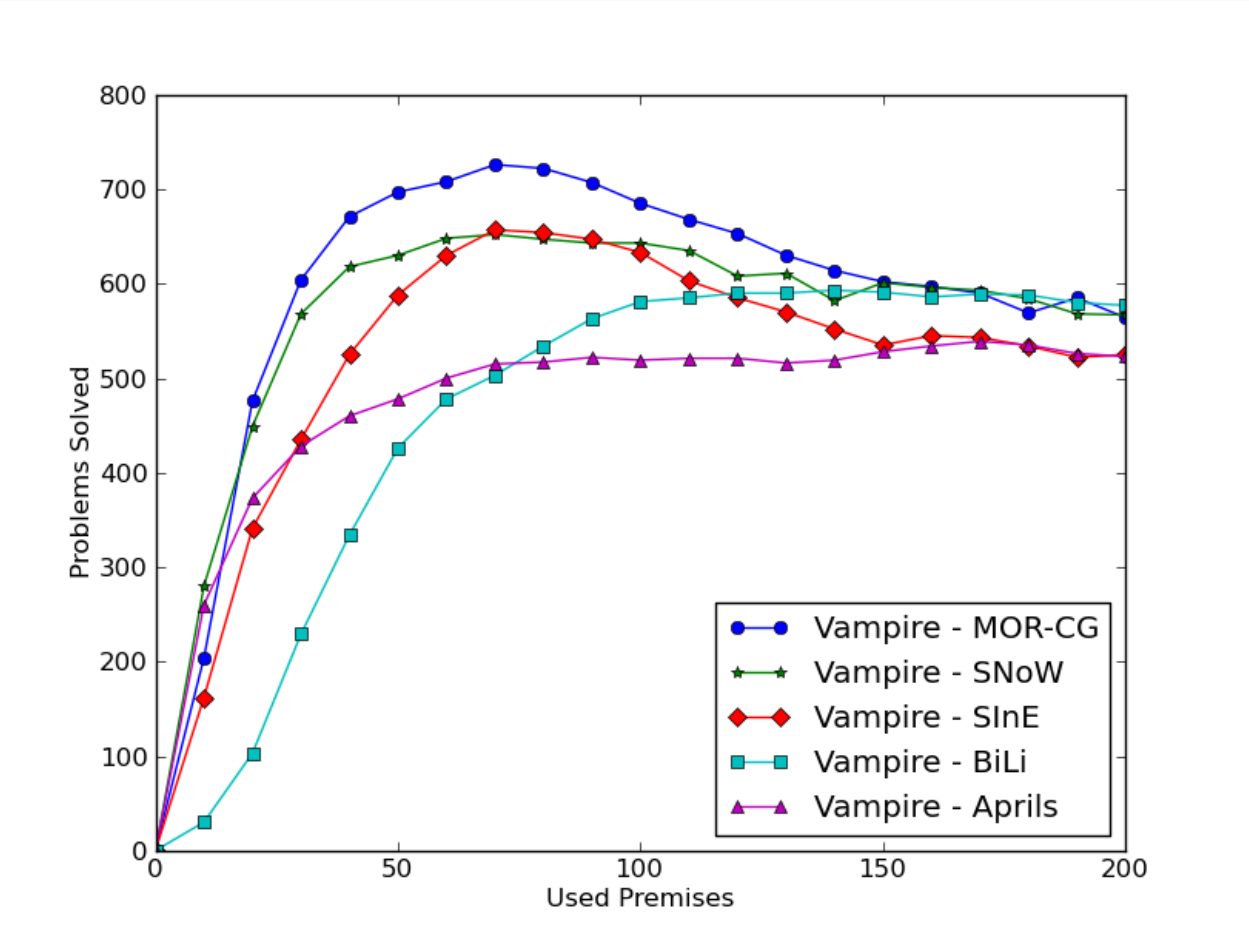
ML Evaluation of methods on MPTP2078 – recall

- Coverage (recall) of facts needed for the Mizar proof in first *n* predictions
- MOR-CG – kernel-based, SNoW - naive Bayes, BiLi - bilinear ranker
- SINe, Aprils - heuristic (non-learning) fact selectors

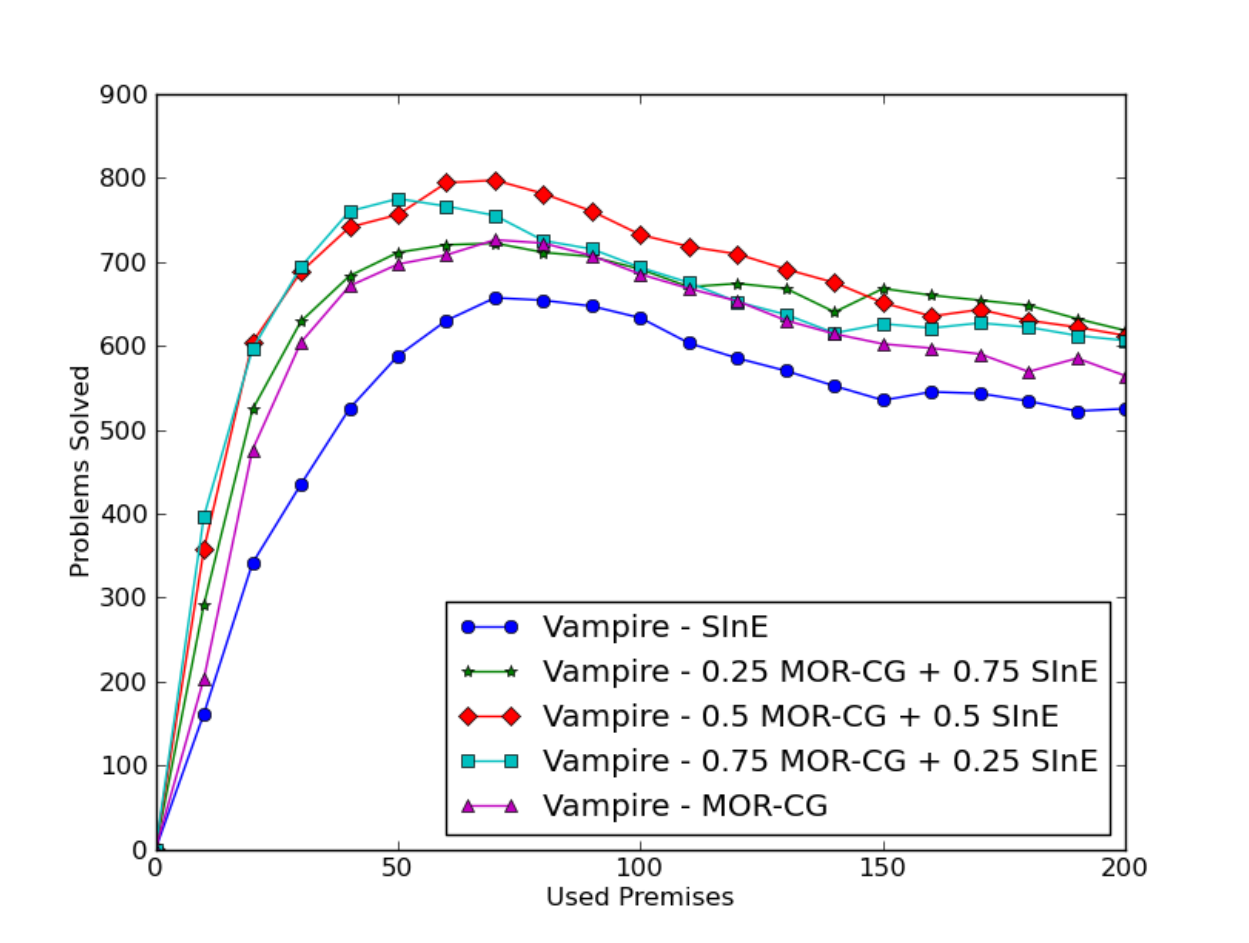


ATP Evaluation of methods on MPTP2078

- Number of the problems proved by ATP when given *n* best-ranked facts
- Good machine learning on previous proofs really matters for ATP!



Combined (ensemble) methods on MPTP2078



Large Evaluation on MML – 60k theorems

14 most covering (40.6%) ML/ATP methods ordered by greedy coverage

Method	Parameters	Prems.	ATP	° -SOTAC	Theorem (%)	Greedy (%)
comb	min_2k_20_20	128	Epar	1728.34	15789 (27.3)	15789 (27.2)
lsi	3200ti_8_80	128	Epar	1753.56	15561 (26.9)	17985 (31.0)
comb	qua_2k_k200_33_33	512	Epar	1520.73	13907 (24.0)	19323 (33.4)
knn	is_40	96	Z3	1634.50	11650 (20.1)	20388 (35.2)
nb	idf010	128	Epar	1630.77	14004 (24.2)	21057 (36.4)
knn	is_80	1024	V	1324.39	12277 (21.2)	21561 (37.2)
geo	r_99	64	V	1357.58	11578 (20.0)	22006 (38.0)
comb	geo_2k_50_50	64	Epar	1724.43	14335 (24.8)	22359 (38.6)
comb	geo_2k_60_20	1024	V	1361.81	12382 (21.4)	22652 (39.1)
comb	har_2k_k200_33_33	256	Epar	1714.06	15410 (26.6)	22910 (39.6)
geo	r_90	256	V	1445.18	13850 (23.9)	23107 (39.9)
lsi	3200ti_8_80	128	V	1621.11	14783 (25.5)	23259 (40.2)
comb	geo_2k_50_00	96	V	1697.10	15139 (26.1)	23393 (40.4)
geo	r_90	256	Epar	1415.48	14093 (24.3)	23478 (40.6)

Summary of Features Used

- From syntactic to more semantic:
- Constant and function symbols
- Walks in the term graph
- Walks in clauses with polarity and variables/skolems unified
- Subterms, de Bruijn normalized
- Subterms, all variables unified
- Matching terms, no generalizations
- terms and (some of) their generalizations
- Substitution tree nodes
- All unifying terms
- Evaluation in a large set of (finite) models
- LSI/PCA combinations of above
- Neural embeddings of above

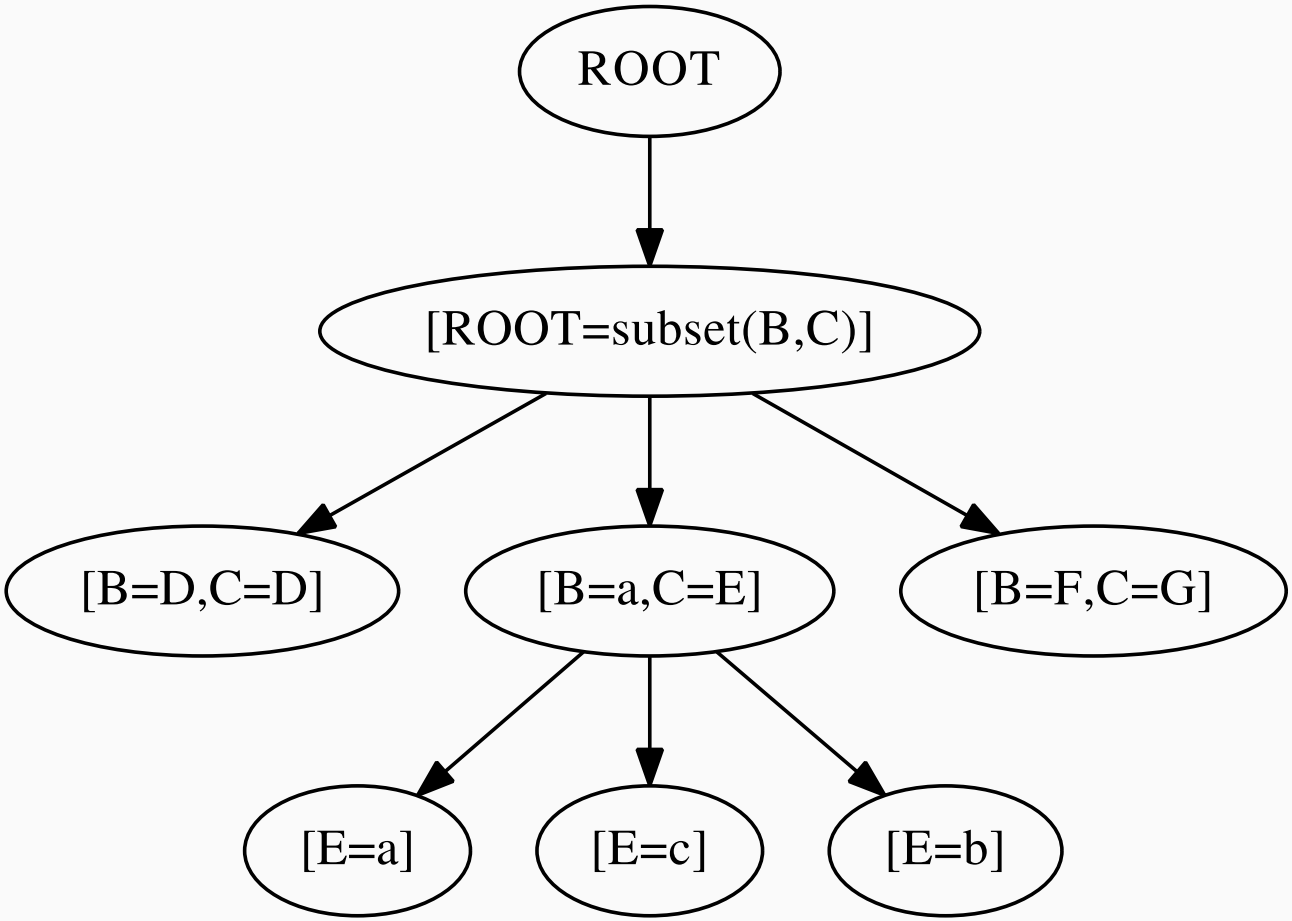
Terms as graphs

Paths in the term

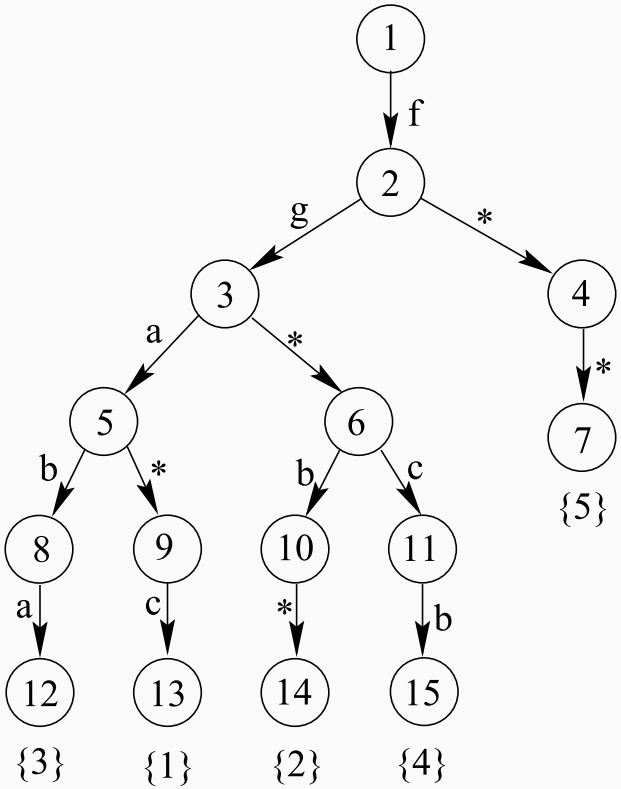
```
f(a,g(b,c),h(d))

f g h  a b c d
f-a f-b f-c f-d f-g f-h g-b g-c h-d
f-g-b f-g-c f-h-d
```

Substitution Trees

```
subset(A,B)
subset(a,b)
subset(a,c)
subset(C,C)
subset(a,a)
```



Discrimination Nets



Generalizations of `f(a,g(b,c),h(d))`

Right:

```
V a b c d h(V) h(d) g(V,V) g(b,V) g(b,c) f(V,V,V)
f(a,V,V) f(a,g(V,V),V) f(a,g(b,V),V) f(a,g(b,c),V)
f(a,g(b,c),h(V)) f(a,g(b,c),h(d))
```

Left:

```
V a b c d h(V) h(d) g(V,V) g(V,c) g(b,c) f(V,V,V)
f(V,V,h(V)) f(V,V,h(d)) f(V,g(V,V),h(d))
f(V,g(V,c),h(d)) f(V,g(b,c),h(d))
```

Positions:

```
V a b c d h(V) h(d) g(V,c) g(b,V) g(b,c)
f(V,g(b,c),h(d)) f(a,V,h(d)) f(a,g(V,c),h(d))
f(a,g(b,V),h(d)) f(a,g(b,c),V) f(a,g(b,c),h(V))
f(a,g(b,c),h(d))
```

Combinations.

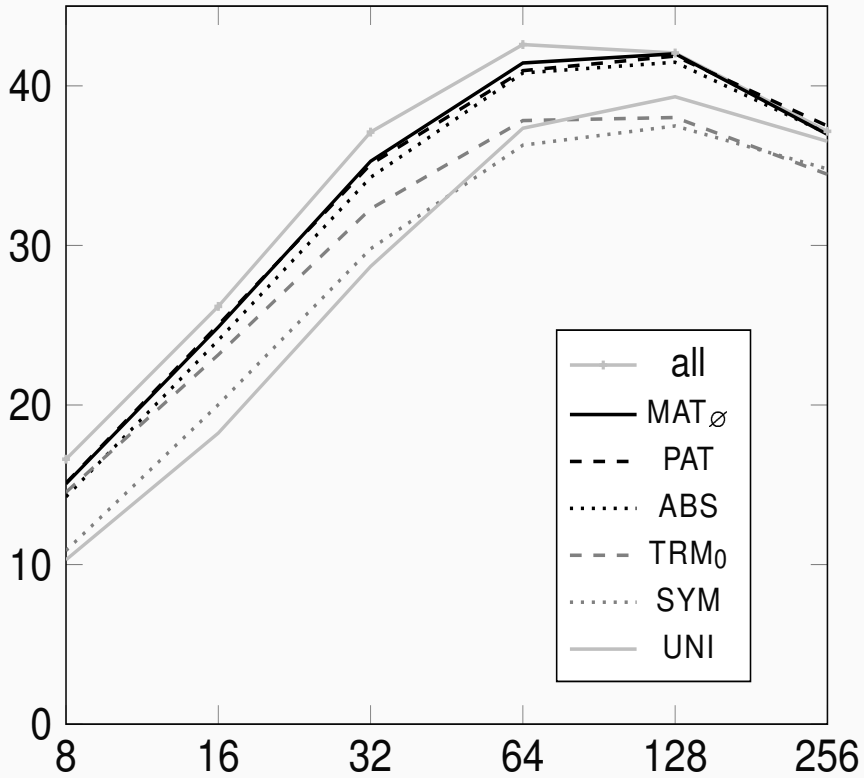
Summary of Features Used

Name	Description
SYM	Constant and function symbols
TRM_0	Subterms, all variables unified
TRM_α	Subterms, de Bruijn normalized
$\mathrm{MAT}_\varnothing$	Matching terms, no generalizations
MAT_r	Repeated gener. of rightmost innermost constant
MAT_l	Repeated gener. of leftmost innermost constant
MAT_1	Gener. of each application argument
MAT_2	Gener. of each application argument pair
$\mathrm{MAT}_\cup$	Union of all above generalizations
PAT	Walks in the term graph
ABS	Substitution tree nodes
UNI	All unifying terms

Feature Statistics (MPTP2078 and MML1147)

Method	Speed (sec)		Number of features		Learning and prediction (sec)	
	MPTP2078	MML1147	total	unique	knn	naive Bayes
SYM	0.25	10.52	30996	2603	0.96	11.80
TRM_α	0.11	12.04	42685	10633	0.96	24.55
TRM_0	0.13	13.31	35446	6621	1.01	16.70
$\text{MAT}_\varnothing$	0.71	38.45	57565	7334	1.49	24.06
MAT_r	1.09	71.21	78594	20455	1.51	39.01
MAT_l	1.22	113.19	75868	17592	1.50	37.47
MAT_1	1.16	98.32	82052	23635	1.55	41.13
MAT_2	5.32	4035.34	158936	80053	1.65	96.41
$\text{MAT}_\cup$	6.31	4062.83	180825	95178	1.71	112.66
PAT	0.34	64.65	118838	16226	2.19	52.56
ABS	11	10800	56691	6360	1.67	23.40
UNI	25	N/A	1543161	6462	21.33	516.24

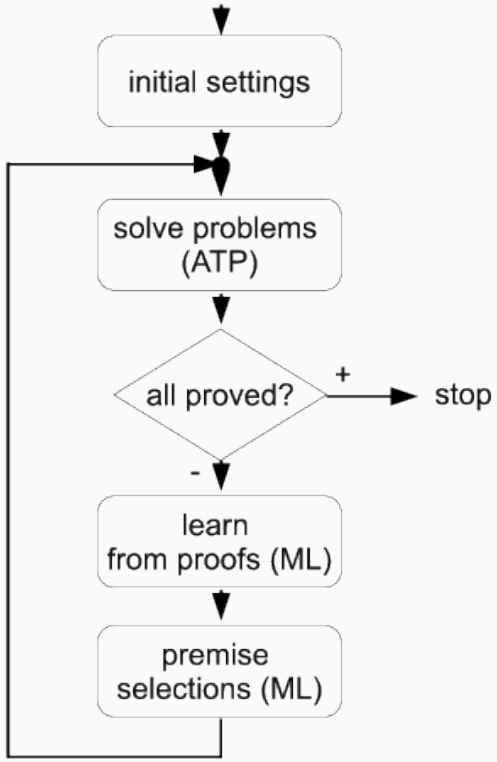
ATP evaluation (E-prover / k-NN)



Method	Proved (%)	Theorems
MAT$_\varnothing$	54.379	1130
MAT$_r$	54.331	1129
MAT$_l$	54.283	1128
PAT	54.235	1127
MAT$_\cup$	53.994	1122
MAT$_1$	53.994	1122
MAT$_2$	53.898	1120
ABS	53.802	1118
TRM$_0$	50.529	1050
UNI	50.241	1044
SYM	48.027	998
TRM$_\alpha$	43.888	912
SYM|TRM$_0$|MAT$_\varnothing$|ABS	55.486	1153

Machine Learner for Automated Reasoning

- MaLARea (2006) – infinite hammering/premise selection
- feedback loop interleaving ATP with learning premise selection
- both syntactic and semantic features for characterizing formulas:
- evolving set of finite (counter)models in which formulas evaluated
- thus the semantic features are evolving as the feedback loop progresses



Combining ML with semantic selection in Malarea

- run model finder before ATPs when it makes sense
- Paradox used when there are less then 64 axioms, and the time limit is small
- detects countersatisfiability much more often and much faster than E and SPASS on small problems
- thousands of (typically different) models usually found in MaLARea runs, creating an interesting database of models relevant for the large theory
- the model database is usable for further purposes

Combining ML with semantic selection in Malarea

- use semantic information for updating axiom relevance
- all formulas from the large theory are evaluated in the models found by model finders
- heuristically, axiom A is more useful for a negated conjecture C if it excludes more models of C
- also, the more rare the exclusion of a certain model of C, the more valuable is the axiom
- let's make invalidity in each model into another feature charactarizing formulas, and use it for machine learning as other features
- this works in the Bayseian framework exactly the same as e.g. symbol-based similarity:
- e.g. an axiom sharing a rare countermodel with a conjecture is promoted in the same way as an axiom sharing a rare symbol with the conjecture

Semantic features in Malarea - Evaluation

- chainy division of the MPTP Challenge: 252 related problems, average size 400 formulas, 1500 formulas in total
- 21 hours overall timelimit
- SRASS: 126, standard MaLARea: 144, with term structure (TS) learning: 149, with TS and semantic guidance (SG): 161
- the base ATPs (E,SPASS): 80 - 90 problems each when 300s for each problem, 104 together (if each 64s for each problem) TS + SG in fast mode: 128 in 2-3 hors, 104 in 30-60 minutes

Some More Improvements and Additions

- Distance-weighted k-nearest neighbor, LSI, boosted trees (XGBoost)
- Matching and transferring concepts and theorems between libraries (Gauthier & Kaliszyk) – allows “superhammers”, conjecturing, and more
- Lemmatization – extracting and considering millions of low-level lemmas
- Neural sequence models, definitional embeddings (Google Research)
- Hammers combined with statistical tactical search: TacticToe (HOL4)
- Learning in binary setting from many alternative proofs (DeepMath,ATPBoost)
- Negative/positive mining (ATPBoost)
- Features of the proof state - syntactic, neural, proof-matching vectors

Matching concepts across libraries

- Same concepts in different proof assistants
  - Problem for proof translation
  - Manually found 7-70 pairs
- Same properties
  - Patterns, like associativity, distributivity ...
  - Same algebraic structures do differ.
- Automatically finds 400 pairs of same concepts
  - In HOL Light, HOL4, Isabelle/HOL
  - Coq: so far only lists analyzed
- Proof advice can be universal?

Representing formulas with binary features

Our current approach is to represent formulas with syntax-based features – symbols, terms, subterms...

Theorem `IRRAT_1:2` again:

```
fof(t2_irrat_1, conjecture,
(?[A]: (v1_xreal_0(A) &
(?[B]: (v1_xreal_0(B) &
(~(v1_rat_1(A)) & (~(v1_rat_1(B)) &
v1_rat_1(k3_power(A, B))))))))).
```

... and its feature description:

```
"v1_xreal_0-V", "v1_xreal_0(A)", "v1_xreal_0",
"v1_rat_1-k3_power", "v1_rat_1-V",
"v1_rat_1(k3_power(A,A))", "v1_rat_1(A)", "v1_rat_1",
"k3_power-V", "k3_power(A,A)", "k3_power"
```

After featuresing the whole MML we obtain 451706 such features.

ATPBoost – Binary settings

There are two possible settings in which we can approach premise selection with machine learning:

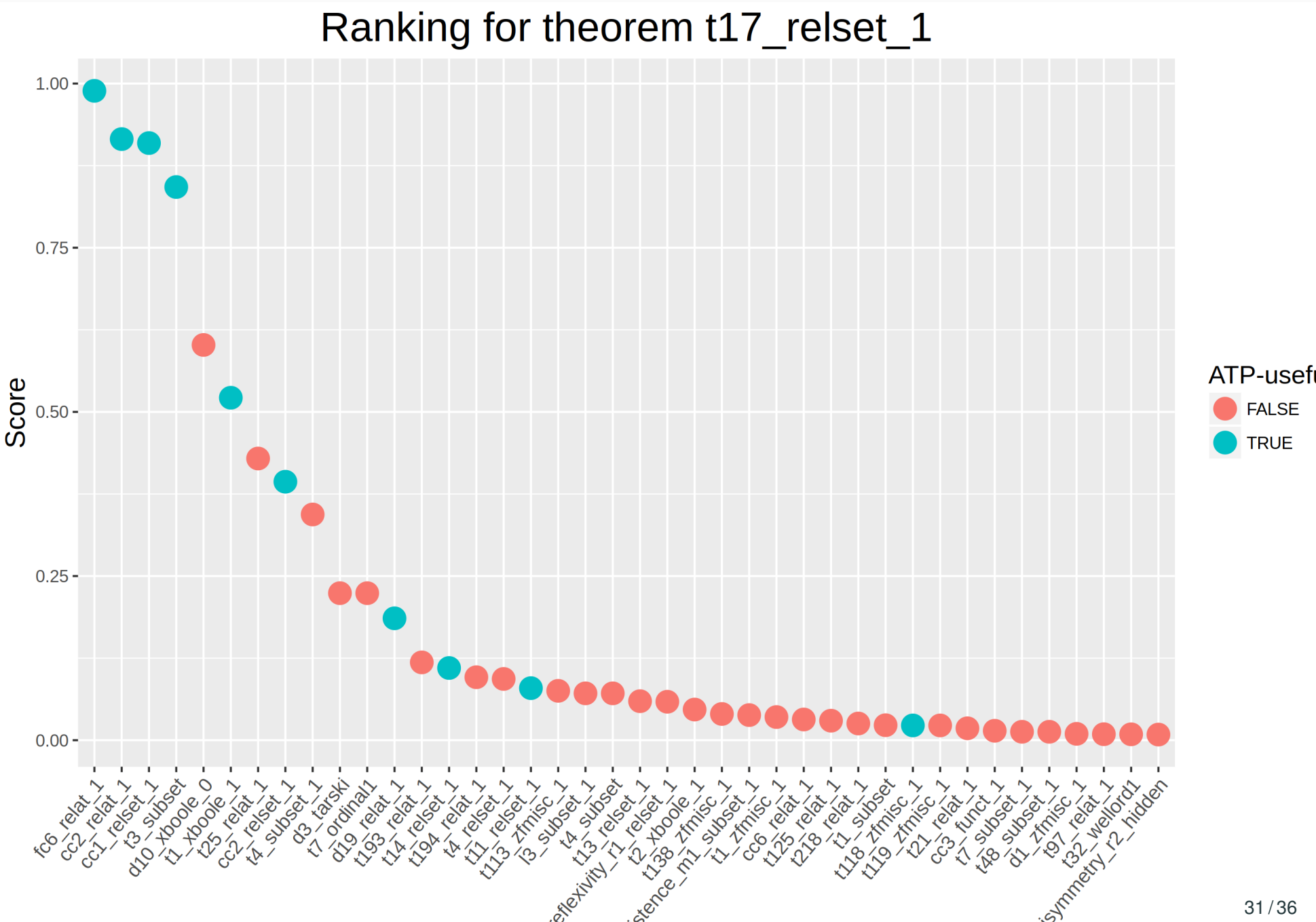
1. *multilabel setting*: here we treat premises used in the proofs as opaque labels on theorems and we train a model capable of labeling conjectures based on their features,
2. *binary setting*: here the aim of the learning model is to recognize pairwise-relevance of the *conjecture-premise* pairs, i.e. to decide what is the chance of the premise being relevant for proving the conjecture based on the features of both the conjecture and the premise.

The first approach is more accessible and was more used so far. The second setting, though, is more general and better for modern, strong ML algorithms.

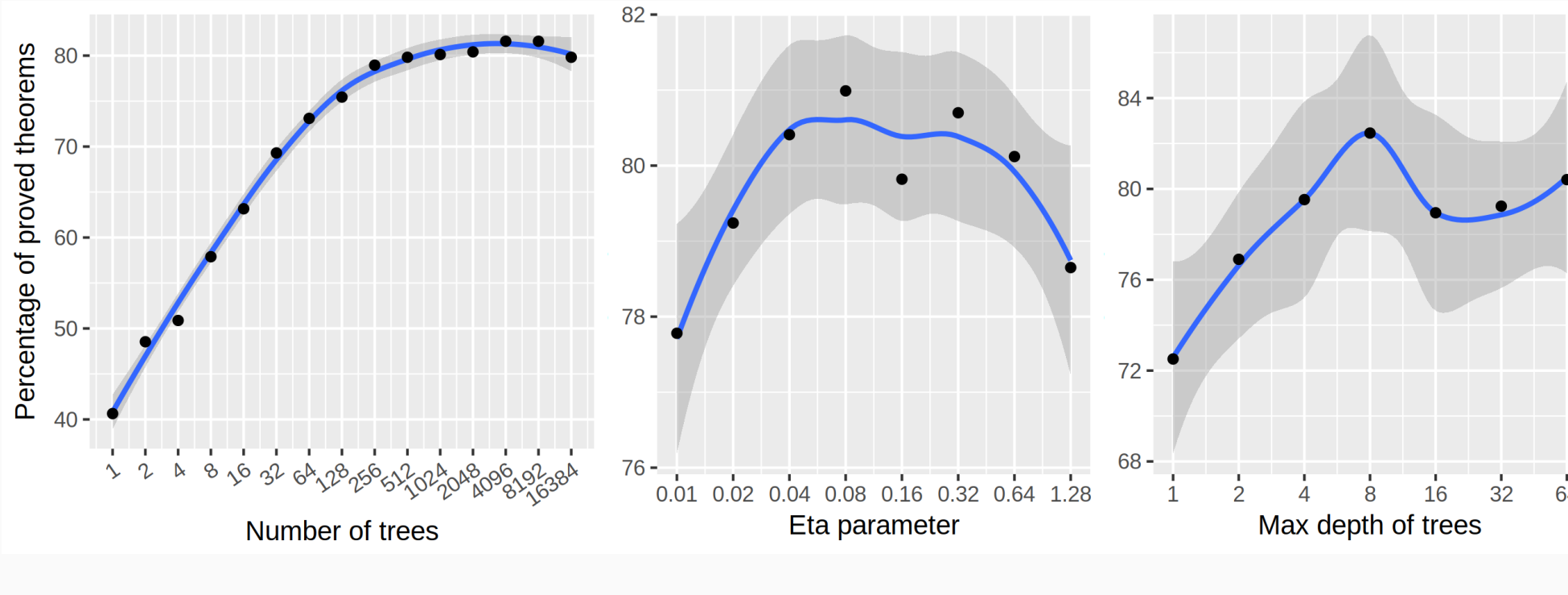
ATPBoost

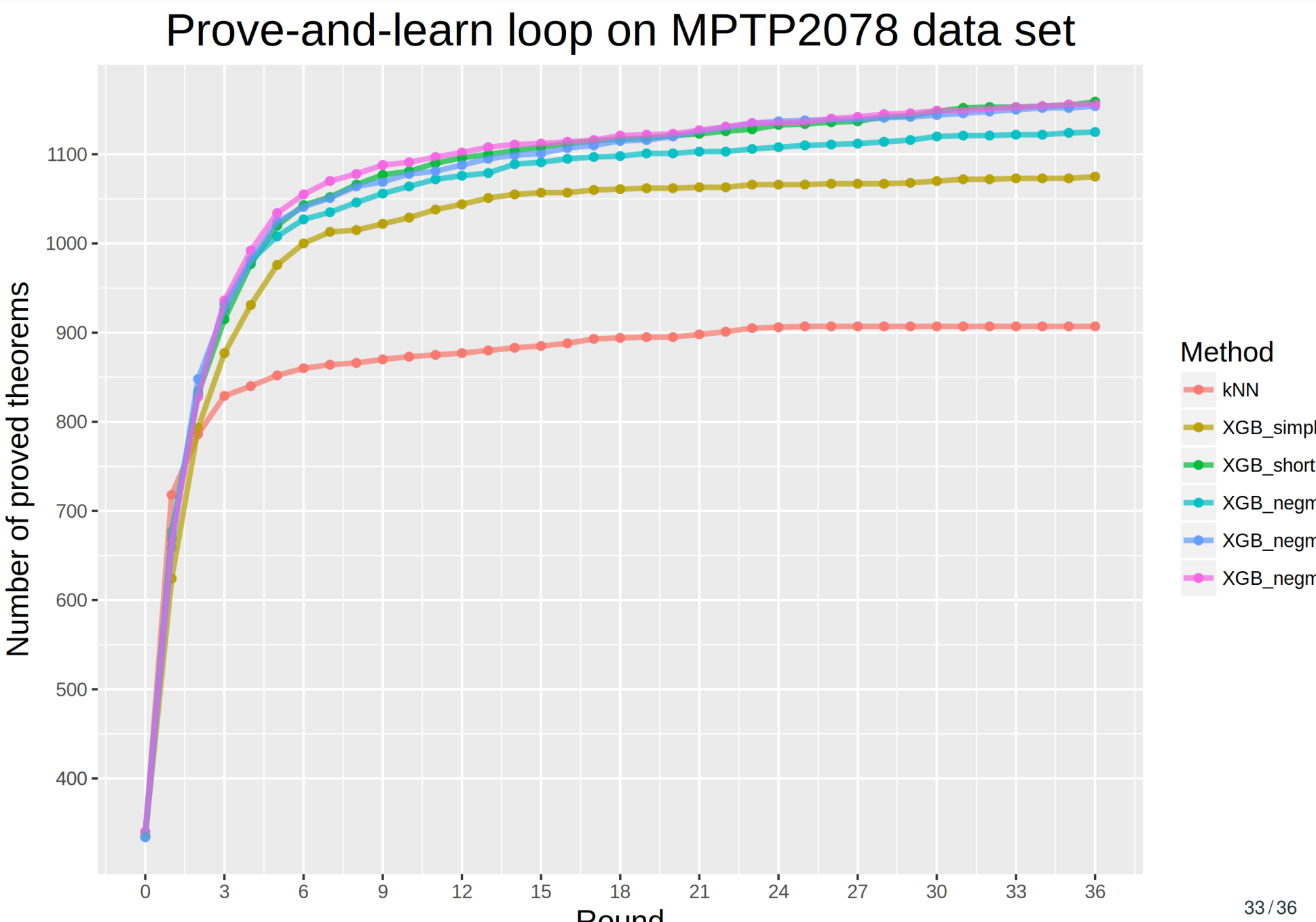
- Positive and negative examples for training set were initially generated from the theorems with proofs in the following way:
  - as positives we take pairs (*theorem-premise*) if premise appears in at least one proof of the *theorem*,
  - negatives are randomly taken from the set of pairs (*theorem-premise*) where the *premise* is available for the *theorem* but there is no ATP-proof of the *theorem* with this *premise*.

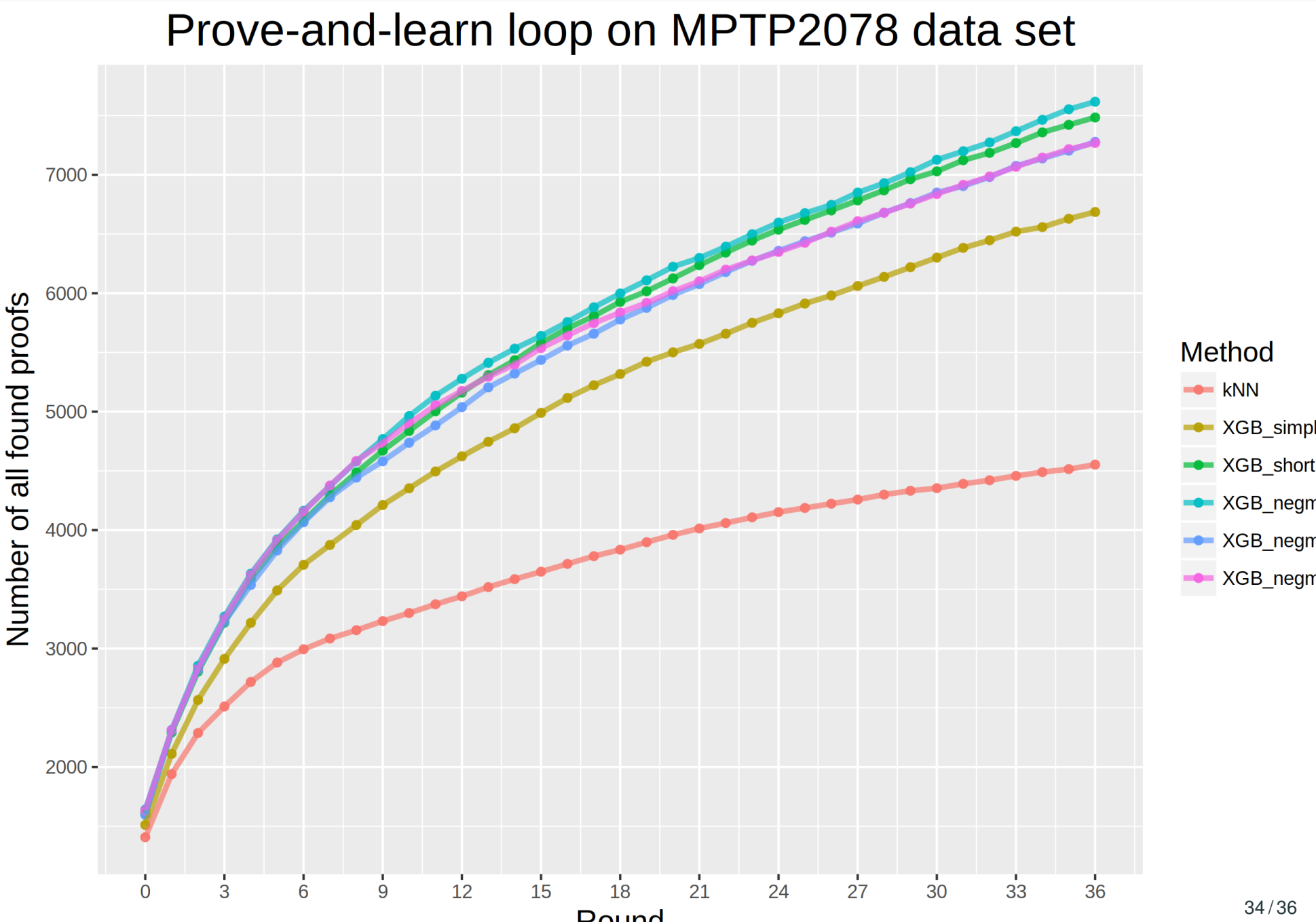
  Every such pair is presented to the ML algorithm as concatenation of its feature representation labeled by 0 or 1.
- After model is trained, we use it to create ranking of premises available for theorem from the outside of training set.

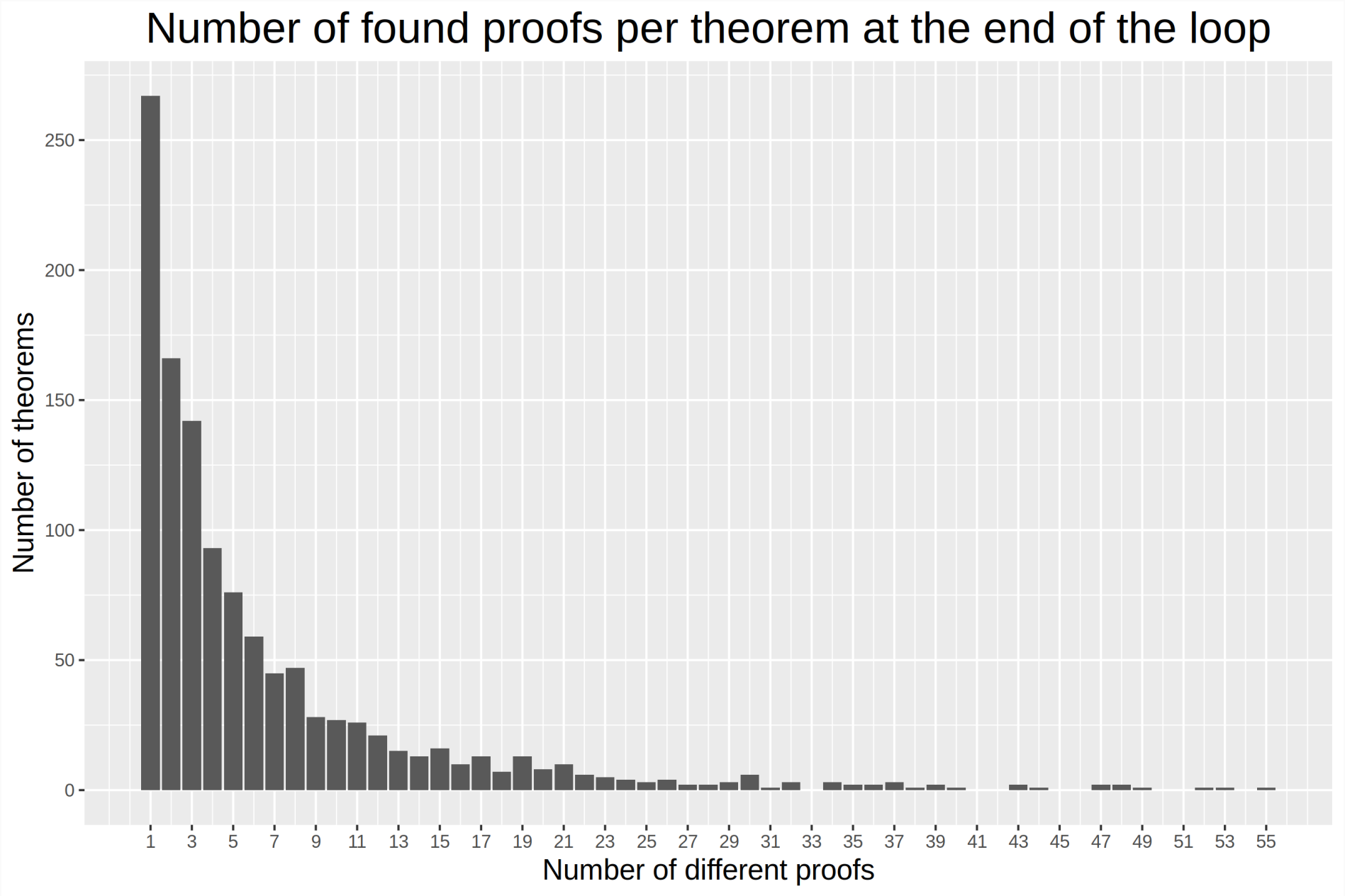

Ranking for theorem t17_relset_1
Score
1.00
0.75
0.50
0.25
0.00
ATP-usefu
FALSE
TRUE
fc6_relat_1
cc2_relat_1
cc1_relset_1
t3_subset
d10_xboole_0
t1_xboole_1
t25_relat_1
cc2_relset_1
t4_subset_1
d3_tarski
t7_ordinal1
d19_relat_1
t193_relat_1
t14_relset_1
t194_relat_1
t4_relset_1
t11_relset_1
t113_zfmisc_1
l3_subset_1
t4_subset
t13_relset_1
eflexivity_r1_relset_1
t2_xboole_1
t138_zfmisc_1
stence_m1_subset_1
t1_zfmisc_1
cc6_relat_1
t125_relat_1
t218_relat_1
t1_subset
t118_zfmisc_1
t119_zfmisc_1
t21_relat_1
cc3_funct_1
t7_subset_1
t48_subset_1
d1_zfmisc_1
t97_relat_1
t32_wellord1
symmetry_r2_hidden

To tune parameters of XGBoost model, training/test split was fixed and each trained model was evaluated with ATP:




Prove-and-learn loop on MPTP2078 data set
Number of proved theorems
1100
1000
900
800
700
600
500
400
0
3
6
9
12
15
18
21
24
27
30
33
36
Round
Method
kNN
XGB_simpl
XGB_short
XGB_negm
XGB_negm
XGB_negm


Prove-and-learn loop on MPTP2078 data set
Number of all found proofs
7000
6000
5000
4000
3000
2000
0
3
6
9
12
15
18
21
24
27
30
33
36
Round
Method
kNN
XGB_simpl
XGB_short
XGB_negm
XGB_negm
XGB_negm


Number of found proofs per theorem at the end of the loop
Number of theorems
250
200
150
100
50
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55
Number of different proofs

Some References

- J. C. Blanchette, C. Kaliszyk, L. C. Paulson, J. Urban: Hammering towards QED. J. Formalized Reasoning 9(1): 101-148 (2016)
- G. Irving, C. Szegedy, A. Alemi, N. Eén, F. Chollet, J. Urban: DeepMath - Deep Sequence Models for Premise Selection. NIPS 2016: 2235-2243
- Bartosz Piotrowski, Josef Urban: ATPboost: Learning Premise Selection in Binary Setting with ATP Feedback. IJCAR 2018: 566-574
- C. Kaliszyk, J. Urban, J. Vyskocil: Efficient Semantic Features for Automated Reasoning over Large Theories. IJCAI 2015: 3084-3090
- Jasmin Christian Blanchette, David Greenaway, Cezary Kaliszyk, Daniel Kühlwein, Josef Urban: A Learning-Based Fact Selector for Isabelle/HOL. J. Autom. Reasoning 57(3): 219-244 (2016)
- Cezary Kaliszyk, Josef Urban: MizAR 40 for Mizar 40. J. Autom. Reasoning 55(3): 245-256 (2015)
- Jesse Alama, Tom Heskes, Daniel Kühlwein, Evgeni Tsivtsivadze, Josef Urban: Premise Selection for Mathematics by Corpus Analysis and Kernel Methods. J. Autom. Reasoning 52(2): 191-213 (2014)
- Cezary Kaliszyk, Josef Urban: Learning-Assisted Automated Reasoning with Flyspeck. J. Autom. Reasoning 53(2): 173-213 (2014)
- J. Urban, G. Sutcliffe, P. Pudlák, J. Vyskocil: MaLARea SG1- Machine Learner for Automated Reasoning with Semantic Guidance. IJCAR 2008: 441-456
- L. Czajka, C. Kaliszyk: Hammer for Coq: Automation for Dependent Type Theory. J. Autom. Reasoning 61(1-4): 423-453 (2018)