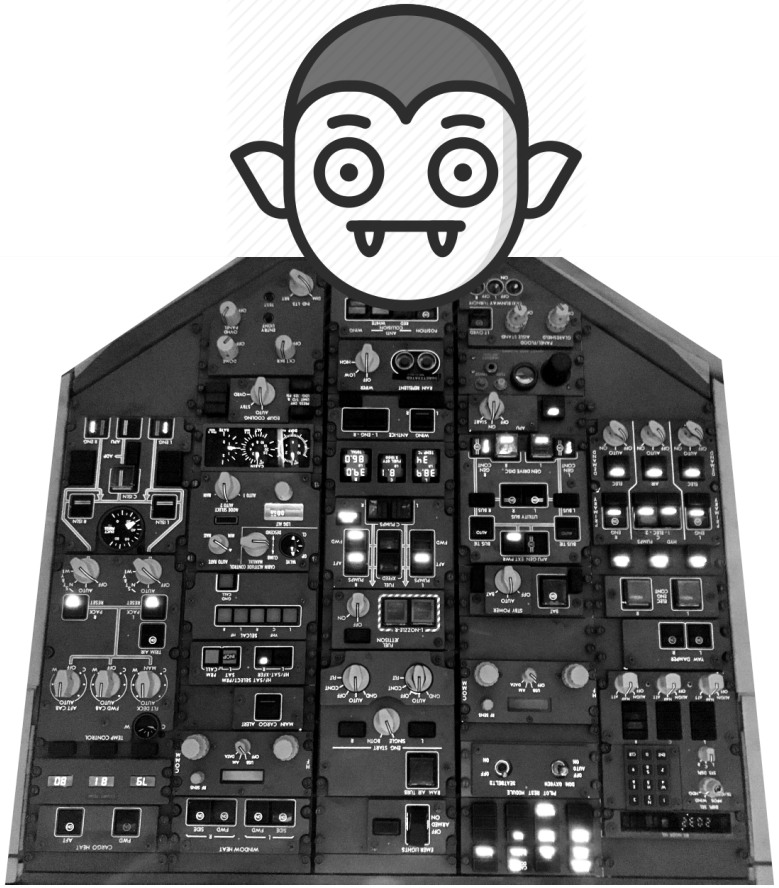
Learning Strategy Design: First Lessons

Martin Suda and Sarah Winkler
Czech Technical University and University of Verona

AITP 2020
16 September 2020

Automated Theorem Proving

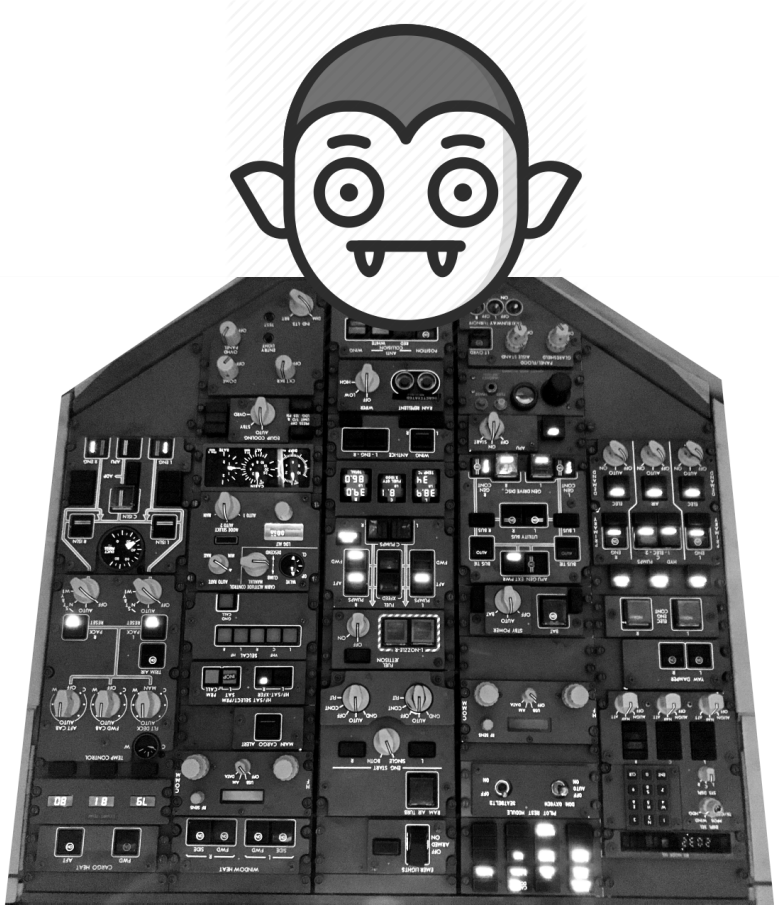
$Q(0,0,0,0)$
$\neg Q(x,y,z,0) \vee Q(x,y,z,1)$
$\neg Q(x,y,0,1) \vee Q(x,y,1,0)$
$\neg Q(x,0,1,1) \vee Q(x,1,0,0)$
$\neg Q(0,1,1,1) \vee Q(1,0,0,0)$
$\neg Q(1,1,1,1)$



SAT
?
UNSAT

Automated Theorem Proving

$$
\begin{aligned}
&Q(0,0,0,0)\\
&\neg Q(x,y,z,0) \vee Q(x,y,z,1)\\
&\neg Q(x,y,0,1) \vee Q(x,y,1,0)\\
&\neg Q(x,0,1,1) \vee Q(x,1,0,0)\\
&\neg Q(0,1,1,1) \vee Q(1,0,0,0)\\
&\neg Q(1,1,1,1)
\end{aligned}
$$



SAT

?

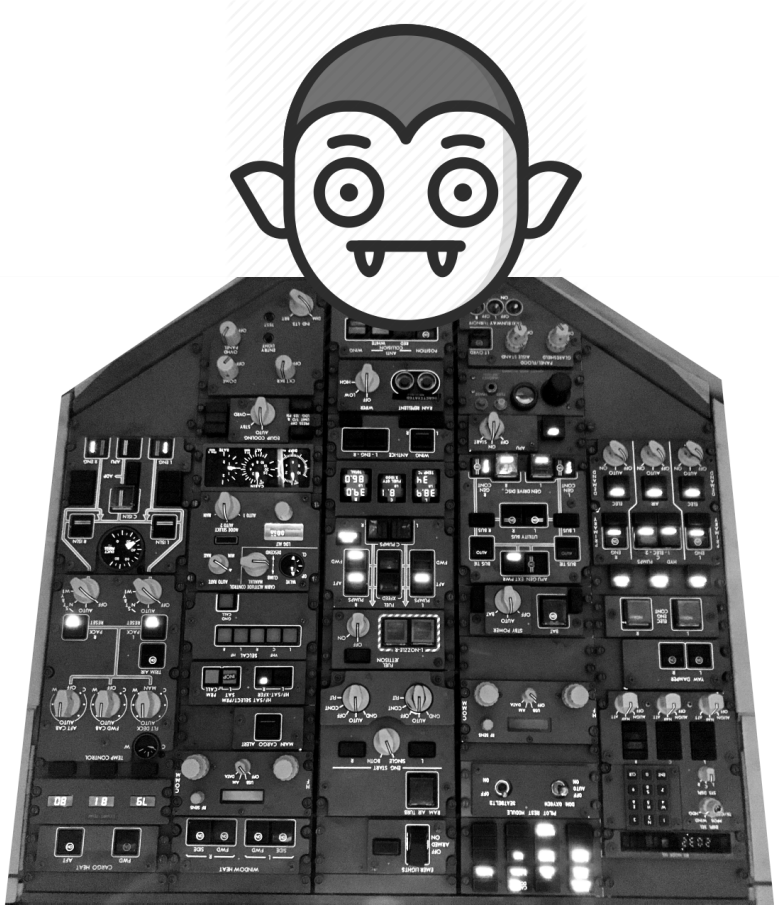
UNSAT

This Talk: Three Experiments

1. predict one of Vampire's 801 CASC strategies for given problem

Automated Theorem Proving

$Q(0,0,0,0)$
$\neg Q(x,y,z,0) \vee Q(x,y,z,1)$
$\neg Q(x,y,0,1) \vee Q(x,y,1,0)$
$\neg Q(x,0,1,1) \vee Q(x,1,0,0)$
$\neg Q(0,1,1,1) \vee Q(1,0,0,0)$
$\neg Q(1,1,1,1)$



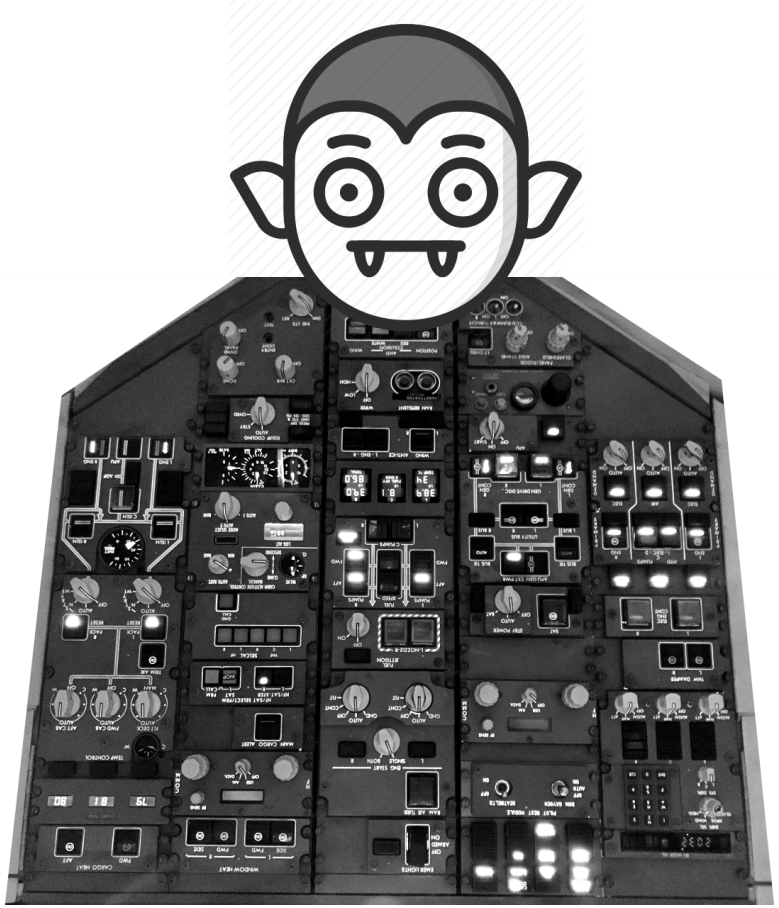
SAT
?
UNSAT

This Talk: Three Experiments

1. predict one of Vampire's 801 CASC strategies for given problem
2. correlate problem features with beneficial strategy components

Automated Theorem Proving

$$
\begin{aligned}
&Q(0,0,0,0)\\
&\neg Q(x,y,z,0) \vee Q(x,y,z,1)\\
&\neg Q(x,y,0,1) \vee Q(x,y,1,0)\\
&\neg Q(x,0,1,1) \vee Q(x,1,0,0)\\
&\neg Q(0,1,1,1) \vee Q(1,0,0,0)\\
&\neg Q(1,1,1,1)
\end{aligned}
$$



SAT

?

UNSAT

This Talk: Three Experiments

1. predict one of Vampire's 801 CASC strategies for given problem
2. correlate problem features with beneficial strategy components
3. correlate problem features with success of CASC tools

Data

Vampire was run for 60sec on all 17574 FOL problems in TPTP 7.2.0 using all 801 strategies $\mathcal{S}$ used in CASC-27

Data

Vampire was run for 60sec on all 17574 FOL problems in TPTP 7.2.0 using all 801 strategies $\mathcal{S}$ used in CASC-27

```
% File     : PUZ015-1 : TPTP v7.2.0. Released v1.0.0.
% Domain   : Puzzles
...
% Source   : [ANL]
% Status   : Satisfiable
% Rating   : 0.89 v7.1.0, 0.88 v7.0.0
% Syntax   : Number of clauses     :   21 (   0 non-Horn;  13 unit;  21 RR)
%            Number of atoms       :   29 (  11 equality)
%            Maximal clause size   :    2 (   1 average)
%            Number of predicates  :    2 (   0 propositional; 2-2 arity)
%            Number of functors    :   16 (  12 constant; 0-8 arity)
%            Number of variables   :   58 (   0 singleton)
%            Maximal term depth    :    3 (   2 average)
%------------------------------------------------------------------------------
cnf(cover_columns_1_and_2,axiom,
    ( ~ achievable(row(X),squares(not_covered,not_covered,Y3,Y4,Y5,Y6,Y7,Y8))
    | achievable(row(X),squares(covered,covered,Y3,Y4,Y5,Y6,Y7,Y8)) )).
cnf(cover_columns_2_and_3,axiom,
    ( ~ achievable(row(X),squares(Y1,not_covered,not_covered,Y4,Y5,Y6,Y7,Y8)) )).
```

Data

Vampire was run for 60sec on all 17574 FOL problems in TPTP 7.2.0 using all 801 strategies $\mathcal{S}$ used in CASC-27

Problem Features

- all 92 problem properties collected by TPTP and Vampire:

  # clauses, # terms, # predicates, # functions, # variables, # connectives, # $\exists$, # $\forall$, # $\vee$, # $\wedge$, # $\neg$, # unit clauses, is EPR, is UEQ, is ground, # Horn clauses, # unit clauses, has sorts, has rationals, has reals, has groups, has rings, has equality, has arrays, has extensionality, max term depth, avg term depth, max predicate arity, avg predicate arity, max function arity, max # variables in clause, ...

Data

Vampire was run for 60sec on all 17574 FOL problems in TPTP 7.2.0 using all 801 strategies $\mathcal{S}$ used in CASC-27

Problem Features

- all 92 problem properties collected by TPTP and Vampire:

  # clauses, # terms, # predicates, # functions, # variables, # connectives, # $\exists$, # $\forall$, # $\vee$, # $\wedge$, # $\neg$, # unit clauses, is EPR, is UEQ, is ground, # Horn clauses, # unit clauses, has sorts, has rationals, has reals, has groups, has rings, has equality, has arrays, has extensionality, max term depth, avg term depth, max predicate arity, avg predicate arity, max function arity, max # variables in clause, ...

- three TPTP features: domain, source, rating

Data

Vampire was run for 60sec on all 17574 FOL problems in TPTP 7.2.0 using all 801 strategies $\mathcal{S}$ used in CASC-27

Problem Features

- all 92 problem properties collected by TPTP and Vampire:

  # clauses, # terms, # predicates, # functions, # variables, # connectives, # $\exists$, # $\forall$, # $\vee$, # $\wedge$, # $\neg$, # unit clauses, is EPR, is UEQ, is ground, # Horn clauses, # unit clauses, has sorts, has rationals, has reals, has groups, has rings, has equality, has arrays, has extensionality, max term depth, avg term depth, max predicate arity, avg predicate arity, max function arity, max # variables in clause, ...

- three TPTP features: domain, source, rating
- three hand-crafted features (approximated):
  - # of unifiable positive and negative literals
  - # terms matching non-variable equation sides
  - # terms unifiable with non-variable equation sides

1. Strategy Prediction

Which strategy works best for a given problem?
Which problem features are decisive?

1. Strategy Prediction

Which strategy works best for a given problem?
Which problem features are decisive?

Task 1

- predict runtime from subset $\mathcal{F}$ of features ("timeout penalty" 300sec)

1. Strategy Prediction

Which strategy works best for a given problem?
Which problem features are decisive?

Task 1

- predict runtime from subset $\mathcal{F}$ of features ("timeout penalty" 300sec)
- random forest regressors
- rating-balanced training and test sets (80% vs 20%)

1. Strategy Prediction

Which strategy works best for a given problem?
Which problem features are decisive?

Task 1

- predict runtime from subset $\mathcal{F}$ of features ("timeout penalty" 300sec)
- random forest regressors
- rating-balanced training and test sets (80% vs 20%)

training phase: for each strategy $s \in \mathcal{S}$ train regressor using features $\mathcal{F}$

test phase: for problem in test set, predict runtime for each strategy, recommend strategy with lowest predicted runtime

1. Strategy Prediction

Which strategy works best for a given problem?
Which problem features are decisive?

Task 1

- predict runtime from subset $\mathcal{F}$ of features ("timeout penalty" 300sec)
- random forest regressors
- rating-balanced training and test sets (80% vs 20%)

training phase: for each strategy $s \in \mathcal{S}$ train regressor using features $\mathcal{F}$

test phase: for problem in test set, predict runtime for each strategy, recommend strategy with lowest predicted runtime

- count how many test problems are solved by recommended strategy

Solved problems

features $\mathcal{F}$	all	no TPTP features	source	# terms	domain	rating	Vampire default single strategy
solved (of 3515)	2583	2548	2342	2180	2241	2166	2013

Solved problems

when predicting from single feature, source works best

features $\mathcal{F}$	all	no TPTP features	source	# terms	domain	rating	Vampire default single strategy
solved (of 3515)	2583	2548	2342	2180	2241	2166	2013

Solved problems

when predicting from single feature, source works best

features $\mathcal{F}$	all	no TPTP features	source	# terms	domain	rating	Vampire default single strategy
solved (of 3515)	2583	2548	2342	2180	2241	2166	2013

Feature importance (without rating)

1. # terms 6%
2. # unifiable pos/neg literals 4.7%
3. # variables 4.2%
4. # atoms 3.8%
5. # connectives 3.5%
6. # functions 3.4%
7. # terms unifiable with equations 3.4%
8. # negations 3.4%
9. # terms matching equations 3.2%
10. # axioms 3.2%
11. # unit clauses 2.9%
12. source 2.8%

Solved problems

when predicting from single feature, source works best

features $\mathcal{F}$	all	no TPTP features	source	# terms	domain	rating	Vampire default single strategy
solved (of 3515)	2583	2548	2342	2180	2241	2166	2013

Interaction matters:
hand-crafted features contribute 11.6%

Feature importance (without r

1. # terms 6%
2. # unifiable pos/neg literals 4.7%
3. # variables 4.2%
4. # atoms 3.8%
5. # connectives 3.5%
6. # functions 3.4%
7. # terms unifiable with equations 3.4%
8. # negations 3.4%
9. # terms matching equations 3.2%
10. # axioms 3.2%
11. # unit clauses 2.9%
12. source 2.8%

Solved problems

when predicting from single feature, source works best

features $\mathcal{F}$	all	no TPTP features	source	# terms	domain	rating	Vampire default single strategy
solved (of 3515)	2583	2548	2342	2180	2241	2166	2013

Interaction matters:
hand-crafted features contribute 11.6%

Feature importance (without r

1. # terms 6%
2. # unifiable pos/neg literals 4.7%
3. # variables 4.2%
4. # atoms 3.8%
5. # connectives 3.5%
6. # functions 3.4%
7. # terms unifi
8. # negations 3
9. # terms matching equations 3.2%
10. # axioms 3.2%
11. # unit clauses 2.9%
12. source 2.8%

Size is important

Solved problems

when predicting from single feature, source works best

features $\mathcal{F}$	all	no TPTP features	source	# terms	domain	rating	Vampire default single strategy
solved (of 3515)	2583	2548	2342	2180	2241	2166	2013

Feature importance (without r

Interaction matters:
hand-crafted features contribute 11.6%

1. # terms 6%
2. # unifiable pos/neg literals 4.7%
3. # variables 4.2%
4. # atoms 3.8%
5. # connectives 3.5%
6. # functions 3.4%
7. # terms unifi
8. # negations 3
9. # terms matching equations 3.2%
10. # axioms 3.2%
11. # unit clauses 2.9%
12. source 2.8%

Size is important

Side remark: regression quality $\neq$ prediction power

- for all features $r^2 = 0.71$, but source-only 0.28 and rating-only 0.41

2. Correlating Problem Features and Parameter Values

Which problem features prefer which parameter values?

2. Correlating Problem Features and Parameter Values

Which problem features prefer which parameter values?

Strategy components

- each strategy consists of set of pairs (o, v) of option o and value v

2. Correlating Problem Features and Parameter Values

Which problem features prefer which parameter values?

Strategy components

- each strategy consists of set of pairs (o, v) of option o and value v

Task 2

compare probability that problem with feature f can be solved by strategy with option o set to a value v to probability that

2. Correlating Problem Features and Parameter Values

Which problem features prefer which parameter values?

Strategy components

- each strategy consists of set of pairs (o, v) of option o and value v

Task 2

compare probability that problem with feature f can be solved by strategy with option o set to a value v to probability that

(a) arbitrary strategy solves problem with feature f (advantage ratio)

2. Correlating Problem Features and Parameter Values

Which problem features prefer which parameter values?

Strategy components

- each strategy consists of set of pairs (o, v) of option o and value v

Task 2

compare probability that problem with feature f can be solved by strategy with option o set to a value v to probability that

(a) arbitrary strategy solves problem with feature f (advantage ratio)

(b) strategy with $o = v$ solves arbitrary problem (surprise ratio)

2. Correlating Problem Features and Parameter Values

Which problem features prefer which parameter values?

Strategy components

- each strategy consists of set of pairs (o, v) of option o and value v

Task 2

compare probability that problem with feature f can be solved by strategy with option o set to a value v to probability that

(a) arbitrary strategy solves problem with feature f (advantage ratio)

(b) strategy with $o = v$ solves arbitrary problem (surprise ratio)

Example

feature	option value	advantage	surprise	#problems
EPR	`age_weight=50`	11%	15%	1512

2. Correlating Problem Features and Parameter Values

Which problem features prefer which parameter values?

Strategy components

- each strategy consists of set of pairs (o, v) of option o and value v

Task 2

compare probability that problem with feature f can be solved by strategy with option o set to a value v to probability that

(a) arbitrary strategy solves problem with feature f (advantage ratio)

(b) strategy with $o = v$ solves arbitrary problem (surprise ratio)

Example

feature	option value	advantage	surprise	#problems
EPR	`age_weight=50`	11%	15%	1512

"strategy s with `age_weight=50` is 11% more likely to solve an EPR problem than an arbitrary strategy, and on EPR s is 15% better than s usually is"

2. Correlating Problem Features and Parameter Values

Which problem features prefer which parameter values?

Strategy components

- each strategy consists of set of pairs (o, v) of option o and value v

Task 2

compare probability that problem with feature f can be solved by strategy with option o set to a value v to probability that

(a) arbitrary strategy solves problem with feature f (advantage ratio)

(b) strategy with $o = v$ solves arbitrary problem (surprise ratio)

Example

feature	option value	advantage	surprise	#problems
EPR	`age_weight=50`	11%	15%	1512

"strategy s with `age_weight=50` is 11% more likely to solve an EPR problem than an arbitrary strategy, and on EPR s is 15% better than s usually is"

 Strongest correlations appear with the source

even for sources with at least 20 problems, 389 correlations where certain option value has $\geqslant$30% advantage on problems from particular source

Strongest correlations appear with the source

even for sources with at least 20 problems, 389 correlations where certain option value has $\geqslant$30% advantage on problems from particular source

source	option value	advantage	surprise	#problems
Col01	`stl=20`	52%	58%	184
Sla93	`sa=fmb`	60%	82%	30
NV07a	`igrr=64/1`	67%	37%	72
WM89	`fmbsr=1.6`	63%	66%	20
Pel09	`age weight=16`	36%	53%	1017
ILTP	`uwa=all`	10%	56%	151

Strongest correlations appear with the source

even for sources with at least 20 problems, 389 correlations where certain option value has $\geqslant$30% advantage on problems from particular source

Correlations identify fragile options

- for options like `stl` and `age_weight`, range is beneficial
- other options are fragile, i.e. only one value works well

Strongest correlations appear with the source

even for sources with at least 20 problems, 389 correlations where certain option value has $\geqslant$30% advantage on problems from particular source

Correlations identify fragile options

- for options like `stl` and `age_weight`, range is beneficial
- other options are fragile, i.e. only

many correlations for EPR and UEQ: saturation algorithm, age-weight limit

Specific correlations

feature	option value	advantage	surprise	#problems
EPR	`age_weight`$\in [50,\ldots,128]$	10%	17%	1512
EPR	`sa=ins`	5%	18%	1512
UEQ	`age_weight=28`	13%	18%	1656
UEQ	`nwc=3`	13%	18%	1656
UEQ	`ins=3`	14%	17%	1656

Strongest correlations appear with the source

even for sources with at least 20 problems, 389 correlations where certain option value has $\geqslant$30% advantage on problems from particular source

Correlations identify fragile options

- for options like `stl` and `age_weight`, range is beneficial
- other options are fragile, i.e. only

many correlations for EPR and UEQ: saturation algorithm, age-weight limit

for UEQ focus on conjecture-derived clauses (by penalizing others)

Specific correlations

feature	option value	advantage	surprise	#problems
EPR	`age_weight`$\in [50,\ldots,128]$	10%	17%	1512
EPR	`sa=ins`	5%	18%	1512
UEQ	`age_weight=28`	13%	18%	1656
UEQ	`nwc=3`	13%	18%	1656
UEQ	`ins=3`	14%	17%	1656

Strongest correlations appear with the source

even for sources with at least 20 problems, 389 correlations where certain option value has $\geqslant$30% advantage on problems from particular source

Correlations identify fragile options

- for options like `stl` and `age_weight`, range is beneficial
- other options are fragile, i.e. only

many correlations for EPR and UEQ: saturation algorithm, age-weight limit

for UEQ focus on conjecture-derived clauses (by penalizing others)

Specific correlations

feature	option value	advantage	surprise	#problems
EPR	`age_weight`$\in [50,\ldots,128]$	10%	17%	1512
EPR	`sa=ins`	5%	18%	1512
UEQ	`age_w`			
UEQ	`nwc=3`			
UEQ	`ins=3`	14%	17%	1656
# cnst > 1556	`age_weight=16`	22%	14%	722
# cnst > 1556	`ep=RST`	12%	21%	722

for many constants, use lighter equational reasoning

Strongest correlations appear with the source

even for sources with at least 20 problems, 389 correlations where certain option value has $\geqslant$30% advantage on problems from particular source

Correlations identify fragile options

- for options like `stl` and `age_weight`, range is beneficial
- other options are fragile, i.e. only

many correlations for EPR and UEQ: saturation algorithm, age-weight limit

for UEQ focus on conjecture-derived clauses (by penalizing others)

for many constants, use lighter equational reasoning

? for many variables, less aggressive limited resource strategy

Specific correlations

feature	option value	advantage	surprise	#problems
EPR	`age_weight`$\in [50,\ldots,128]$	10%	17%	1512
EPR	`sa=ins`	5%	18%	1512
UEQ	`age_w`			
UEQ	`nwc=3`			
UEQ				
# cnst > 1556				
# cnst > 1556	`ep=RST`	12%	21%	722
vars/clause > 183	`stl`$\in \{150, 210\}$	20%	10%	178

Strongest correlations appear with the source

even for sources with at least 20 problems, 389 correlations where certain option value has $\geqslant$30% advantage on problems from particular source

Correlations identify fragile options

- for options like `stl` and `age_weight`, range is beneficial
- other options are fragile, i.e. only

many correlations for EPR and UEQ: saturation algorithm, age-weight limit

for UEQ focus on conjecture-derived clauses (by penalizing others)

? for UEQ, eager inequality splitting

for many constants, use lighter equational reasoning

? for many variables, less aggressive limited resource strategy

Specific correlations

feature	option value	advantage	surprise	#problems
EPR	`age_weight`$\in [50,\ldots,128]$			
EPR	`sa=ins`	5%	18%	1512
UEQ	`age_w`			
UEQ	`nwc=3`			
UEQ				
# cnst $>$ 1556				
# cnst $>$ 1556	`ep=RST`	12%	21%	722
vars/clause $>$ 183	`stl`$\in \{150, 210\}$	20%	10%	178

3. Correlating Problem Features and CASC Tools

Given a problem, which tool works best?

3. Correlating Problem Features and CASC Tools

Given a problem, which tool works best?

Task 3

compare probability that problem with feature f can be solved by tool t to

(a) probability that other tool solves problems with feature f

3. Correlating Problem Features and CASC Tools

Given a problem, which tool works best?

Task 3

compare probability that problem with feature f can be solved by tool t to

(a) probability that other tool solves problems with feature f

(b) probability that tool t solves arbitrary problem (overperformance)

3. Correlating Problem Features and CASC Tools

Given a problem, which tool works best?

Task 3

compare probability that problem with feature f can be solved by tool t to

(a) probability that other tool solves problems with feature f

(b) probability that tool t solves arbitrary problem (overperformance)

Classes where Vampire does not work best

feature	# problems	tool
`has_reals`	279	CVC4 1.7
`has_interpreted_equality`	869	CVC4 1.7
$>$ 54 positive axioms	1120	Leo III 1.3
source `Hoe08/Sta08`	441/140	versions of E

3. Correlating Problem Features and CASC Tools

Given a problem, which tool works best?

Task 3

compare probability that problem with feature f can be solved by tool t to

(a) probability that other tool solves problems with feature f

(b) probability that tool t solves arbitrary problem (overperformance)

Classes where Vampire does not work best

feature	# problems	tool
`has_reals`	279	CVC4 1.7
`has_interpreted_equality`	869	CVC4 1.7
$>$ 54 positive axioms	1120	Leo III 1.3
source `Hoe08/Sta08`	441/140	versions of E

Overperformance

iProver, Z3, Zipperposition on `EPR`, versions of E on UEQ, ...

Summary

- some number crunching to find correlations between problem features and successful strategies/strategy properties

Summary

- some number crunching to find correlations between problem features and successful strategies/strategy properties
- can predict reasonably good strategy out of fixed set
- identify influential and relevant features: size, interference (but TPTP characteristics like source highly significant)

Summary

- some number crunching to find correlations between problem features and successful strategies/strategy properties
- can predict reasonably good strategy out of fixed set
- identify influential and relevant features: size, interference (but TPTP characteristics like source highly significant)
- identify problem clusters where other tools than Vampire prevail

Summary

- some number crunching to find correlations between problem features and successful strategies/strategy properties
- can predict reasonably good strategy out of fixed set
- identify influential and relevant features: size, interference (but TPTP characteristics like source highly significant)
- identify problem clusters where other tools than Vampire prevail
- data and collection of TPTP problem features is available

  `http://cl-informatik.uibk.ac.at/users/swinkler/learn_strat/`

Summary

- some number crunching to find correlations between problem features and successful strategies/strategy properties
- can predict reasonably good strategy out of fixed set
- identify influential and relevant features: size, interference (but TPTP characteristics like source highly significant)
- identify problem clusters where other tools than Vampire prevail
- data and collection of TPTP problem features is available

  `http://cl-informatik.uibk.ac.at/users/swinkler/learn_strat/`

Future investigations

- correlations for multiple features
- play with dimensionality reduction
- use such analysis to build good strategy schedules
- suggestions?