Learning to Learn in Interactive Constraint Acquisition

Dimos Tsouros
dimos.tsouros@kuleuven.be

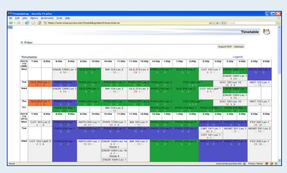
Senne Berden
senne.berden@kuleuven.be

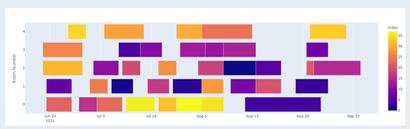
Tias Guns
tias.guns@kuleuven.be

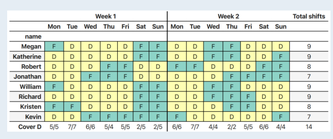
KU LEUVEN
Computer Science, DTAI

Motivation

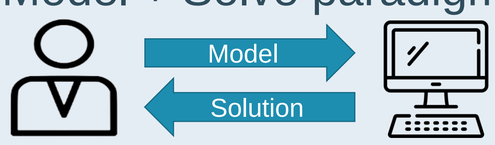
- Constraint programming (CP)
  - Solve combinatorial problems in various domains





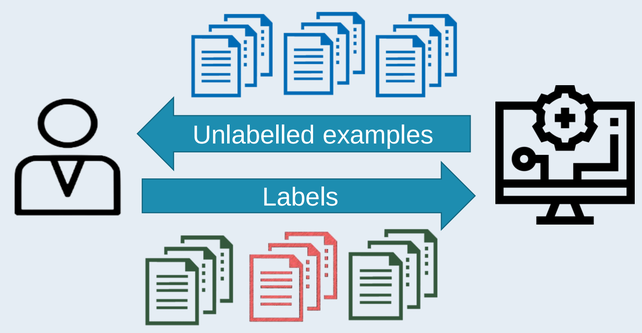


- Model + Solve paradigm

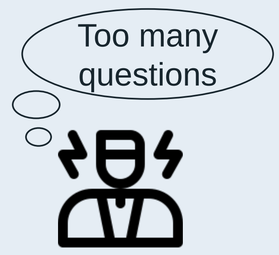


- Constraint Acquisition (CA)

- Interactive CA



- Main Challenge:

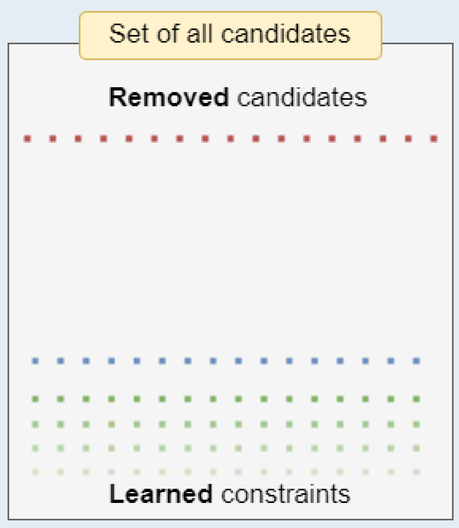


Our contribution:
First time using statistical ML to guide interactive CA, learning to learn during the acquisition process

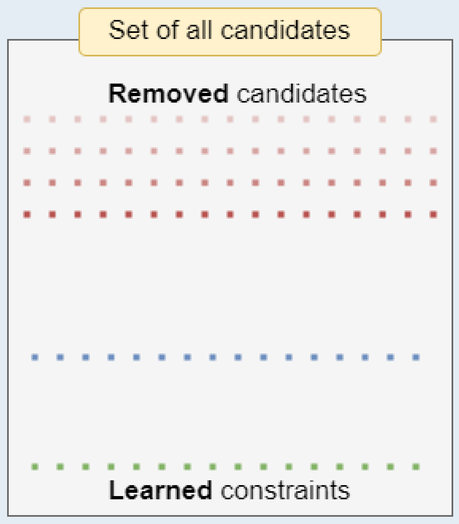
Background

Adapting Candidate Elimination

Using a set of candidate constraints: $B = \{ x_1 = x_2, x_1 \neq x_3, x_3 \neq x_5, x_1 > x_2 \dots \}$



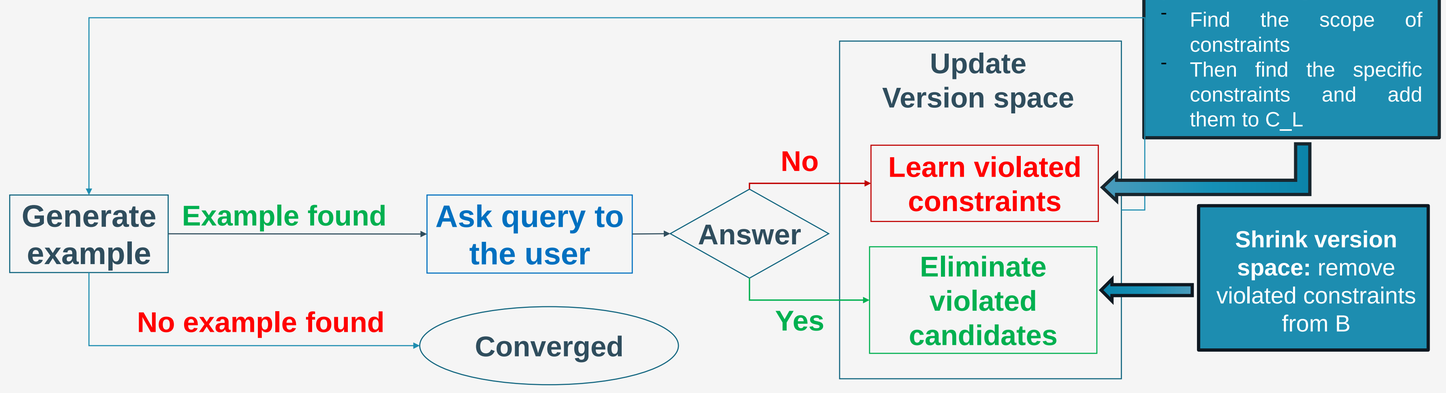
Negative examples



Positive examples

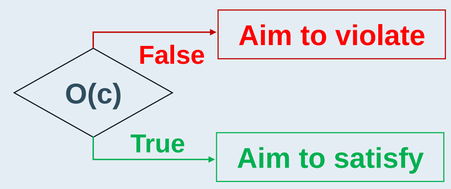
Getting as input a set of candidate constraints B, find a set of constraints $C_L \subseteq B$ *s.t.* $sol(C_L) = sol(C_T)$*, with* C_T *being the target constraint set*

Interactive Constraint Acquisition Template



Guiding Query Generation

- *Use of* ***Oracle O(C)*** $\rightarrow$ *True if* $P(c \in CT \mid c \in B)$ is high enough



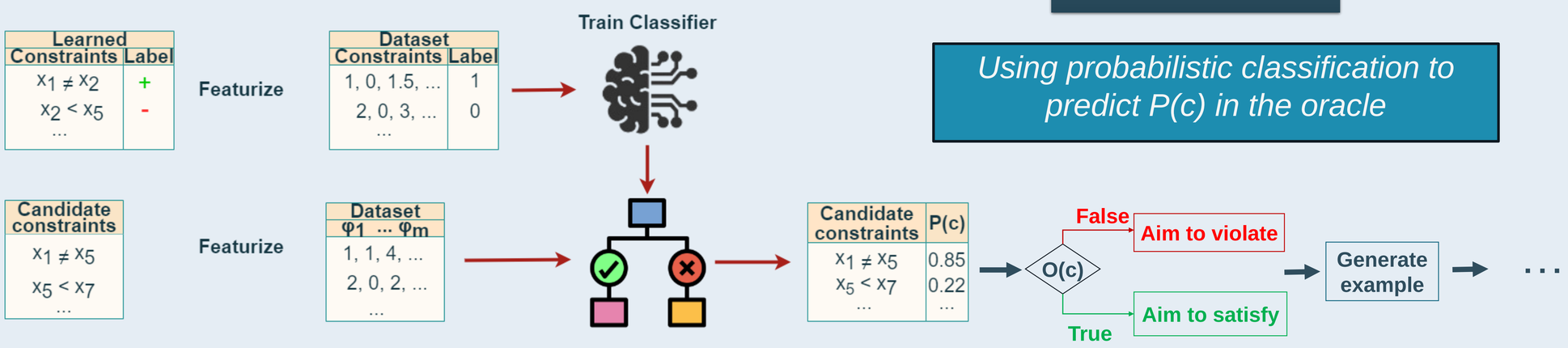
Objective function:

$$e = \underset{e \in Sol(CL \wedge B)}{\operatorname{argmax}} \sum_{c \in B} [\![e \notin sol(c)]\!] \cdot (1 - |\Gamma| \cdot [\![O(c)]\!])$$

Better probabilistic estimate P(c) leads to better generated queries

Contribution

Using statistical ML to learn patterns in constraints



Using probabilistic classification to predict P(c) in the oracle

Statistical ML learns the structure implicitly, query-based learning makes it explicit

We also extended guidance to all queries of CA:

Learn violated constraints — This is a 2-step process: **Find a minimal conflicting scope** → **Find the constraint**

We showed how to adjust the objective function to use in these 2 steps

Feature Representation of Constraints

ID	Name	Type
1	Relation	Categ.
2	Arity	Int
3	Has_constant	Bool
4	Constant	Int
5	Var_name_same	Bool
6	Var_Ndims_same	Bool
7	Var_Ndims_max	Int
8	Var_Ndims_min	Int
9	Var_dim$_i$_has	Bool
10	Var_dim$_i$_same	Bool
11	Var_dim$_i$_max	Int
12	Var_dim$_i$_min	Int
13	Var_dim$_i$_avg	Float
14	Var_dim$_i$_spread	Float

Evaluation

Benchmarks:
- Sudoku 9x9
- Exam Timetabling
- Nurse Rostering

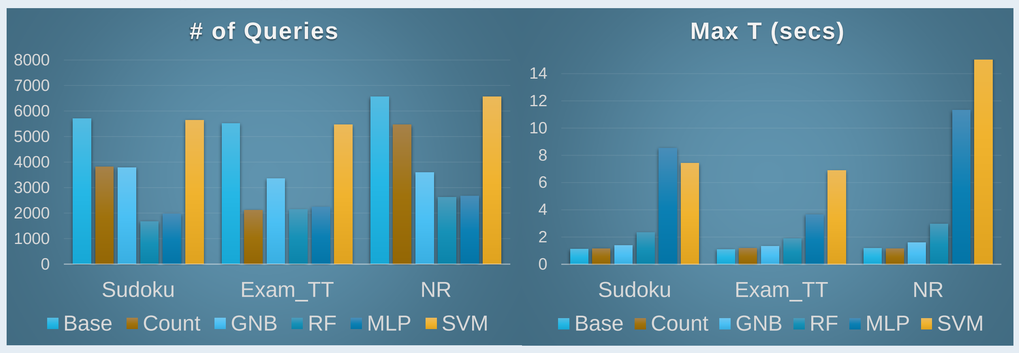
Classifiers:
- Random Forests (RF)
- Gaussian Naïve Bayes (GNB)
- Multilayer Perceptron (MLP)
- Support Vector Machines (SVM)
- A frequentist counting method (Count)

Metrics used:
- # of Queries: Total number of queries until convergence
- Max T (secs): The maximum waiting time of the user in seconds

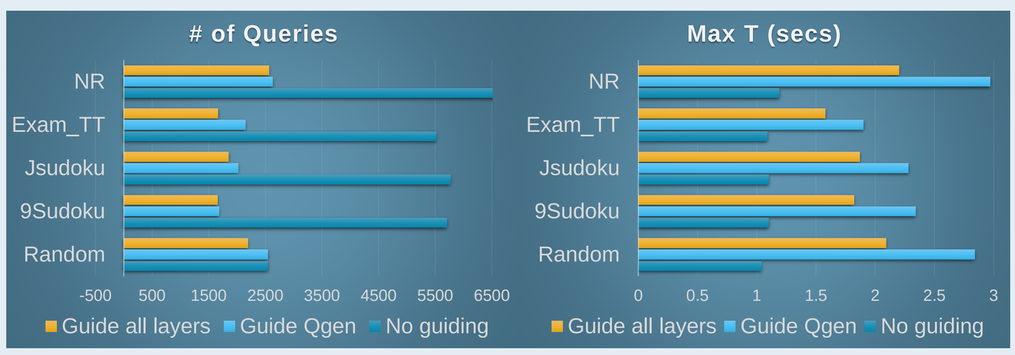
Total number of queries reduced up to 70%

Results

- Guiding query generation using different classifiers



- Guiding all layers using RF



Conclusions

- **Statistical ML can detect patterns in (incomplete) constraint models and can be used successfully to generalize and guide Interactive CA towards better queries.**
- **Total number of queries reduced up to 70%**

This research received funding from
- the European Research Council (ERC) under the EU Horizon 2020 research and innovation programme (Grant No 101002802, CHAT-Opt)
- the EU Horizon 2020 research and innovation programme (Grant No 101070149, project Tuples).