Machine Learning for Mathematical Software

Matthew England
Coventry University

Machine Learning for Mathematical Software Session

International Congress on Mathematical Software
ICMS 2018
24-27 July 2018
University of Notre Dame, IN, USA

Speaker funded by EU Project 712689 (SC^2) and EPSRC Project EP/R019622/1.

Outline

1. Introduction
2. Machine Learning for CAD (Survey of Author's Work)
    - CAD
    - Preconditioning with Groebner Bases
    - Choosing a Variable Ordering
3. Machine Learning for other Mathematical Software (Potentials and Inspirations)
    - ML elsewhere in Computer Algebra?
    - ML for SAT/SMT
    - Other ML for MS Success Stories

Machine Learning (1/20)

Machine Learning: tools that use statistical techniques to give computer systems the ability to *learn* rules from data; i.e. improve their performance on a specific task without changing their explicit programming.

Machine Learning

Machine Learning: tools that use statistical techniques to give computer systems the ability to *learn* rules from data; i.e. improve their performance on a specific task without changing their explicit programming.

Found great success in recent years driven by the advances in both computer hardware and the availability of data. E.g.

- Google's AlphaGo beating professional human Go player!
- Cambridge Analytica (illegally) mining Facebook Data to target advertising and influence elections!

Lower profile but equally impressive applications in huge variety of industries.

Machine Learning (1/20)

Machine Learning: tools that use statistical techniques to give computer systems the ability to *learn* rules from data; i.e. improve their performance on a specific task without changing their explicit programming.

Found great success in recent years driven by the advances in both computer hardware and the availability of data. E.g.

- Google's AlphaGo beating professional human Go player!
- Cambridge Analytica (illegally) mining Facebook Data to target advertising and influence elections!

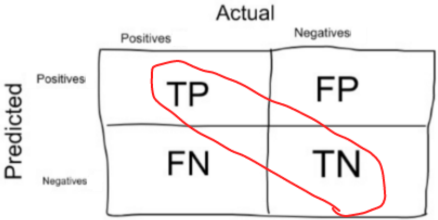
Lower profile but equally impressive applications in huge variety of industries.

Should Mathematical Software be jumping on the bandwagon?

Machine Learning for Software

Machine learning is becoming more common place in the software development process, e.g. in testing and security analysis.

But Machine Learning is inherently probabilistic: the most common metric to evaluate its use is **Accuracy**, something most mathematical software would not sacrifice!



$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Choices, switches, orderings, etc.

But as most developers know, mathematical software often comes with choices which, while having **no affect on the correctness** of the end result, could have a **big affect on the resources required** to find it.

Choices, switches, orderings, etc.

But as most developers know, mathematical software often comes with choices which, while having **no affect on the correctness** of the end result, could have a **big affect on the resources required** to find it.

- In what order to perform a search that may terminate early?
- Which set of competing exact algorithms to use for this problem instance?

Often based on man-made heuristics or "*magic constants*".

Decisions where the underlying relationships are not understood, but are not themselves the key object of study $\Longrightarrow$ good candidate for machine learning.

Choices, switches, orderings, etc. (3/20)

But as most developers know, mathematical software often comes with choices which, while having **no affect on the correctness** of the end result, could have a **big affect on the resources required** to find it.

- In what order to perform a search that may terminate early?
- Which set of competing exact algorithms to use for this problem instance?

Often based on man-made heuristics or "*magic constants*".

Decisions where the underlying relationships are not understood, but are not themselves the key object of study $\Longrightarrow$ good candidate for machine learning.

Potential even for machine learning to give insight on new mathematical results?

Machine Learning for Mathematical Software (4/20)

Our ICMS session will explore these issues in a variety of mathematical software:

- Computer Algebra Systems
- Satisfiability Checking Solvers
- Automated Reasoning
- Mathematical Knowledge Management

Some currently make more use of machine learning than others.
Aims:

- Learn from best practice
- Inspiration for new applications

This Talk

1. Introduction
2. Machine Learning for CAD (Survey of Author's Work)
    - CAD
    - Preconditioning with Groebner Bases
    - Choosing a Variable Ordering
3. Machine Learning for other Mathematical Software (Potentials and Inspirations)
    - ML elsewhere in Computer Algebra?
    - ML for SAT/SMT
    - Other ML for MS Success Stories

Outline

1. Introduction
2. Machine Learning for CAD (Survey of Author's Work)
   - CAD
   - Preconditioning with Groebner Bases
   - Choosing a Variable Ordering
3. Machine Learning for other Mathematical Software (Potentials and Inspirations)
   - ML elsewhere in Computer Algebra?
   - ML for SAT/SMT
   - Other ML for MS Success Stories

Topic: Real Quantifier Elimination

Quantifier Elimination

Given: Quantified formulae in prenex form with atoms integral polynomial constraints.

Produce: Logically equivalent quantifier free formula.

Topic: Real Quantifier Elimination

Quantifier Elimination

Given: Quantified formulae in prenex form with atoms integral polynomial constraints.
Produce: Logically equivalent quantifier free formula.

For example:

Given: $\exists x, x^2 + bx + 1 \leq 0$

Produce: $(b \leq -2) \vee (b \geq 2)$

Topic: Real Quantifier Elimination

Quantifier Elimination

Given: Quantified formulae in prenex form with atoms integral polynomial constraints.
Produce: Logically equivalent quantifier free formula.

For example:

Given: $\exists x, x^2 + bx + 1 \leq 0$

Produce: $(b \leq -2) \vee (b \geq 2)$

Cylindrical Algebraic Decomposition (CAD) is the only implemented complete algorithm that can perform real quantifier elimination. Usually (but not always) implemented in Computer Algebra Systems.

Cylindrical Algebraic Decomposition

CAD works by building a:

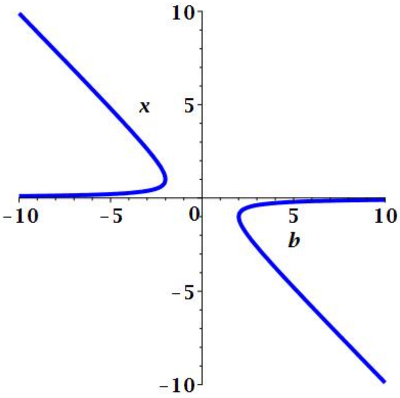
- **Decomposition** of $\mathbb{R}^n$ such that the polynomials involved in the input have constant sign (+/0/−) in each cell, and thus the formula constant truth value.
- The cells are **semi-algebraic** meaning they are described by finite number of polynomial constraints.
- The cells are **cylindrical** meaning projection is trivial from the cell description, and projections of any two cells are identical or disjoint.

Thus existential QE via projection of true cells onto free variables.

QE via CAD Example

Recall from earlier the problem:

$$\exists x, x^2 + bx + 1 \leq 0$$



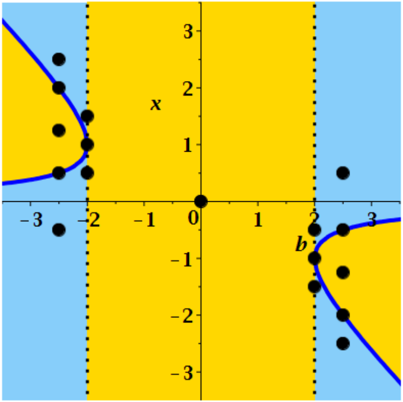
QE via CAD Example

Recall from earlier the problem:

$$\exists x, x^2 + bx + 1 \leq 0$$

To solve we:

Build a sign-invariant CAD for $f = x^2 + bx + 1$.



QE via CAD Example

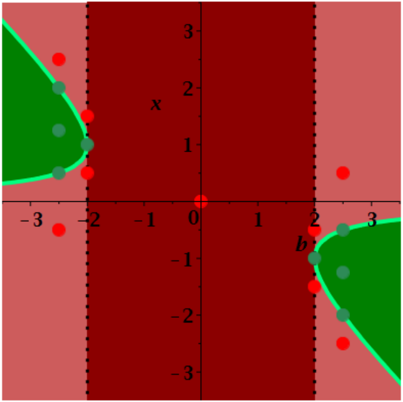
Recall from earlier the problem:

$$\exists x, x^2 + bx + 1 \leq 0$$

To solve we:

Build a sign-invariant CAD for $f = x^2 + bx + 1$.

Tag each cell true or false according to $f \leq 0$.



QE via CAD Example

Recall from earlier the problem:

$$\exists x, x^2 + bx + 1 \leq 0$$

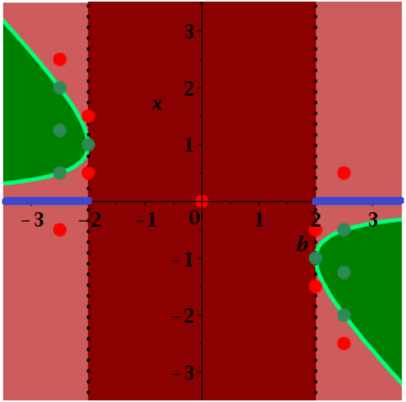
To solve we:

Build a sign-invariant CAD for $f = x^2 + bx + 1$.

Tag each cell true or false according to $f \leq 0$.

Take disjunction of projections of true cells:

$$b < -2 \vee b = -2 \vee b = 2 \vee b > -2$$



Where are the choices?

We produce the CAD using exact arithmetic (potentially algebraic numbers) to ensure correctness.

But there are a variety of choices a developer or user must make:

- Variable Ordering
- Input pre-processing
- Designation of equational constraints
- Ordering of constraints

$\vdots$

Where are the choices?

We produce the CAD using exact arithmetic (potentially algebraic numbers) to ensure correctness.

But there are a variety of choices a developer or user must make:

- Variable Ordering
- Input pre-processing
- Designation of equational constraints
- Ordering of constraints

  ⋮

These do not affect the correctness of CAD (it will have requested invariance property) but can affect:

- Coarseness of the CAD (more cell divisions than needed to provide that invariance property).
- Time take to produce CAD.

Decision: GB Preconditioning

Let $E = \{e_1, e_2, \ldots\}$ be a set of polynomials;
$G = \{g_1, g_2, \ldots\}$ be a Gröbner Basis for E;
and B be any Boolean combination of constraints.
Then

$$\Phi = (e_1 = 0 \wedge e_2 = 0 \wedge \ldots) \wedge B \text{ and}$$
$$\Psi = (g_1 = 0 \wedge g_2 = 0 \wedge \ldots) \wedge B$$

are logically equivalent.

Changing Φ to Ψ is a common pre-conditioning for CAD input. It is usually beneficial, but there are examples where it makes it a lot worse! Human made heuristic to predict this not particularly accurate.

ML for GB Decision

Z. Huang, M. England, J.H. Davenport, L.C. Paulson.
Using Machine Learning to decide when to Precondition Cylindrical Algebraic Decomposition with Groebner Bases.
Proc. 18th Intl. Sym. on Symbolic & Numeric Algorithms for Scientific Computing (SYNASC '16), pp. 45–52. IEEE, 2016.

Dataset of 1000 problems with multiple equations: 75% easier for CAD after GB was taken.

Trained a Support Vector Machine (SVM) classifier with radial basis function to make the decision.

Problem features were simple algebraic properties of polynomial system (degrees, occurrence of variables etc.)

ML for GB Decision

Z. Huang, M. England, J.H. Davenport, L.C. Paulson.
Using Machine Learning to decide when to Precondition Cylindrical Algebraic Decomposition with Groebner Bases.
Proc. 18th Intl. Sym. on Symbolic & Numeric Algorithms for Scientific Computing (SYNASC '16), pp. 45–52. IEEE, 2016.

Dataset of 1000 problems with multiple equations: 75% easier for CAD after GB was taken.

Trained a Support Vector Machine (SVM) classifier with radial basis function to make the decision.

Problem features were simple algebraic properties of polynomial system (degrees, occurrence of variables etc.)

- Needed features of GB for classifier to make good decisions.
- OK, since for any problem where CAD is tractable GB is trivial.
- Feature selection experiments improved accuracy.

Decision: CAD Variable Ordering

CADs are defined with respect to an ordering on variables (cylindricity, projection etc.) For QE one must order variables as they are quantified; but there is no restriction on free variables and adjacent quantifiers of the same type may be swapped.

Well known that this can dramatically affect the feasibility of a problem. In fact, there are a class of problems in which one variable ordering gives output of double exponential complexity in the number of variables and another output of a constant size!

There are several heuristics published on how to make the choice but each can be misled.

ML for CAD Variable Ordering Choice

Z. Huang, M. England, D. Wilson, J.H. Davenport, L.C. Paulson, J. Bridge.
Applying machine learning to the problem of choosing a heuristic to select the variable ordering for cylindrical algebraic decomposition.
Intelligent Computer Mathematics (LNCS 8543), pp. 92–107. Springer, 2014.

Choice is not binary: potentially n! orderings! Instead, learn which of of three man-made heuristics to trust for a problem instance.

Experiments on 7000 problems identified substantial subclasses on which each of the three made the best decision.

Trained three SVMs and used relative magnitude of their margin values to pick which heuristic to follow. Machine learned choice did significantly better than any one heuristic.

Outline

1. Introduction
2. Machine Learning for CAD (Survey of Author's Work)
    - CAD
    - Preconditioning with Groebner Bases
    - Choosing a Variable Ordering
3. Machine Learning for other Mathematical Software (Potentials and Inspirations)
    - ML elsewhere in Computer Algebra?
    - ML for SAT/SMT
    - Other ML for MS Success Stories

ML for Other Choices in CAD / QE

Only other (known) work on this is by the SyNRAC team on the Todai Robot Project for ordering subformulae to tackle with QE (**Next Talk!**)

ML for Other Choices in CAD / QE

Only other (known) work on this is by the SyNRAC team on the Todai Robot Project for ordering subformulae to tackle with QE (**Next Talk!**)

There are certainly other decisions to be made for CAD / QE. Perhaps to the user the one of most importance is which QE implementation to use for a problem!

EPSRC project EP/R019622/1 is just starting on these topics.

Job Advert

Postdoc opportunity at Coventry on *Embedding Machine Learning into Quantifier Elimination Procedures* (deadline 12th August):

`https://www.jobs.ac.uk/job/BLF769/`

ML for Other Choices in Computer Algebra

- How and when to simplify mathematical expressions?

```
> (x^2-1)/(x-1);
```

$$\frac{x^2-1}{-1+x}$$

```
> simplify(%);
```

$$1+x$$

```
> (x^100 -1)/(x-1);
```

$$\frac{x^{100}-1}{-1+x}$$

```
> simplify(%);
```

$$(1+x)\left(x^2+1\right)\left(x^4+x^3+x^2+x+1\right)\left(x^4-x^3+x^2-x+1\right)\left(x^8-x^6+x^4-x^2+1\right)\left(x^{20}+x^{15}+x^{10}+x^5+1\right)\left(x^{20}-x^{15}+x^{10}-x^5+1\right)\left(x^{40}-x^{30}+x^{20}-x^{10}+1\right)$$

ML for Other Choices in Computer Algebra (14/20)

- How and when to simplify mathematical expressions?
- Which algorithm to use for given well defined task, e.g. symbolic integration?

Learn from features of input?

```
> (x^2-1)/(x-1);
```

$$\frac{x^2-1}{-1+x}$$

```
> simplify(%);
```

$$1+x$$

```
> (x^100 -1)/(x-1);
```

$$\frac{x^{100}-1}{-1+x}$$

```
> simplify(%);
```

$$(1+x)\left(x^2+1\right)\left(x^4+x^3+x^2+x+1\right)\left(x^4-x^3+x^2-x+1\right)\left(x^8-x^6+x^4-x^2+1\right)\left(x^{20}+x^{15}+x^{10}+x^5+1\right)\left(x^{20}-x^{15}+x^{10}-x^5+1\right)\left(x^{40}-x^{30}+x^{20}-x^{10}+1\right)$$

ML for Other Choices in Computer Algebra

- How and when to simplify mathematical expressions?
- Which algorithm to use for given well defined task, e.g. symbolic integration?
- For a not-well defined task (e.g. Maple's `solve`).

Learn from features of input?

Features of session history?

```
> (x^2-1)/(x-1);
```

$$\frac{x^2-1}{-1+x}$$

```
> simplify(%);
```

$$1+x$$

```
> (x^100 -1)/(x-1);
```

$$\frac{x^{100}-1}{-1+x}$$

```
> simplify(%);
```

$$(1+x)\left(x^2+1\right)\left(x^4+x^3+x^2+x+1\right)\left(x^4-x^3+x^2-x+1\right)\left(x^8-x^6+x^4-x^2+1\right)\left(x^{20}+x^{15}+x^{10}+x^5+1\right)\left(x^{20}-x^{15}+x^{10}-x^5+1\right)\left(x^{40}-x^{30}+x^{20}-x^{10}+1\right)$$

ML for Other Choices in Computer Algebra

- How and when to simplify mathematical expressions?
- Which algorithm to use for given well defined task, e.g. symbolic integration?
- For a not-well defined task (e.g. Maple's `solve`).

Learn from features of input?
Features of session history?

See upcoming talks in this session from developers of MAPLE and REDLOG and talk this afternoon on ML for group theory.

```
> (x^2-1)/(x-1);
```

$$\frac{x^2-1}{-1+x}$$

```
> simplify(%);
```

$$1+x$$

```
> (x^100 -1)/(x-1);
```

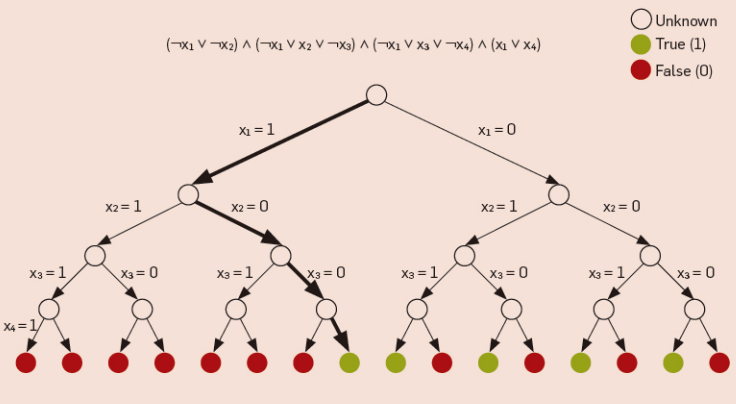
$$\frac{x^{100}-1}{-1+x}$$

```
> simplify(%);
```

$$(1+x)\left(x^2+1\right)\left(x^4+x^3+x^2+x+1\right)\left(x^4-x^3+x^2-x+1\right)\left(x^8-x^6+x^4-x^2+1\right)\left(x^{20}+x^{15}+x^{10}+x^5+1\right)\left(x^{20}-x^{15}+x^{10}-x^5+1\right)\left(x^{40}-x^{30}+x^{20}-x^{10}+1\right)$$

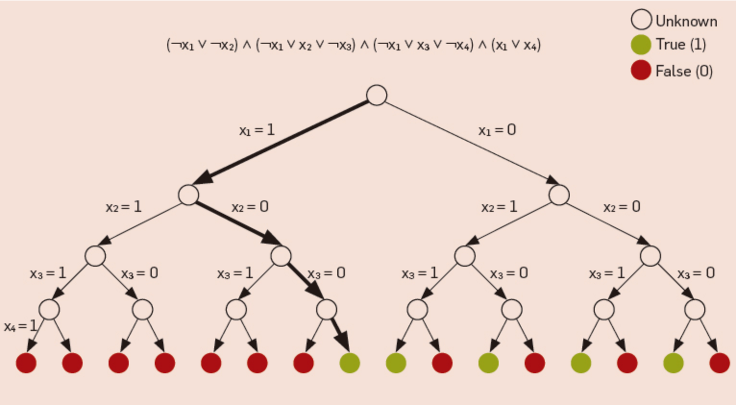
SAT Solvers

SAT-solvers: tools dedicated to solving the Boolean SAT problem.



SAT Solvers

SAT-solvers: tools dedicated to solving the Boolean SAT problem.



The CDCL algorithm uses a structured search with propagation & clause addition to avoid similar bad guesses.

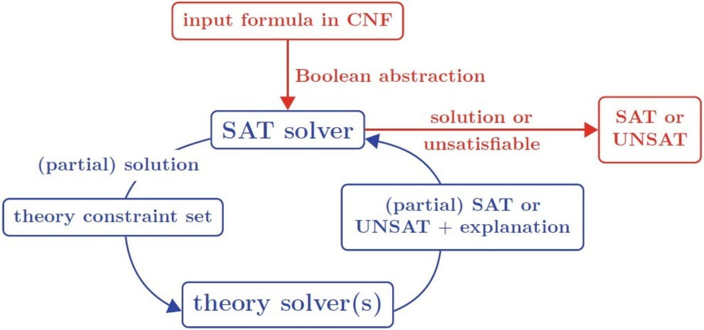
ML in SAT-Solvers

Quite a few examples of ML to improve SAT-Solvers:

- The portfolio solver SATZILLA takes sets of problem instances and solvers, and constructs a portfolio solver optimizing a given objective function (such as mean runtime, percent of instances solved, or score in a competition).
- The MAPLESAT solver views the question of branching as an optimisation problem (solved with ML) where the objective is to maximize the learning rate, defined as the propensity for variables to generate learnt clauses.
- Choice of initial value to variable allocation often chosen randomly but can instead be set with a Monte-Carlo approach.

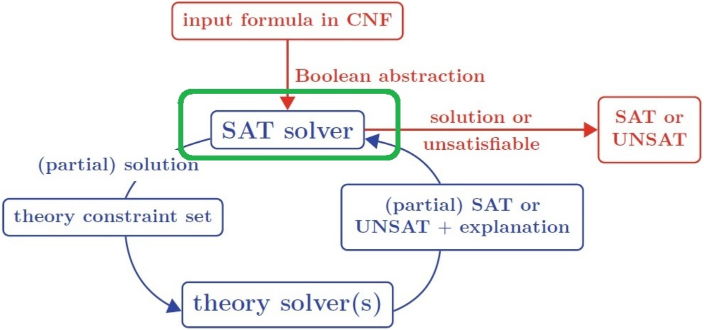
SMT Solvers

Satisfiability Module Theories (SMT): iteratively find solutions to the Boolean skeleton of problem with SAT solver, then query with a theory solver, potentially learning new clauses.



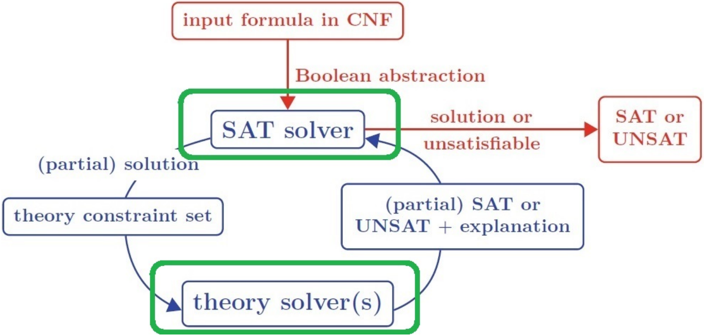
SMT Solvers

Satisfiability Module Theories (SMT): iteratively find solutions to the Boolean skeleton of problem with SAT solver, then query with a theory solver, potentially learning new clauses.



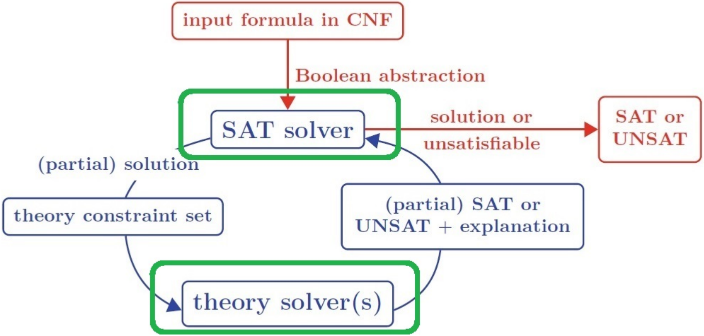
SMT Solvers

Satisfiability Module Theories (SMT): iteratively find solutions to the Boolean skeleton of problem with SAT solver, then query with a theory solver, potentially learning new clauses.



SMT Solvers

Satisfiability Module Theories (SMT): iteratively find solutions to the Boolean skeleton of problem with SAT solver, then query with a theory solver, potentially learning new clauses.



See talk this afternoon on potential for ML in SMT.

Mathematical Knowledge Management (18/20)

Area of great potential as many tasks here involve Natural Language Processing, a well studied application for machine learning, although care has to be taken in adapting for mathematics language.

Tasks which can be tackled with ML:

- Automated key phrase extraction.
- Mathematics handwriting recognition.
- Automated classification of papers via the Mathematics Subject Classification (MSC) system.

Mathematical Knowledge Management

Area of great potential as many tasks here involve Natural Language Processing, a well studied application for machine learning, although care has to be taken in adapting for mathematics language.

Tasks which can be tackled with ML:

- Automated key phrase extraction.
- Mathematics handwriting recognition.
- Automated classification of papers via the Mathematics Subject Classification (MSC) system.
  See talk this afternoon!

Automated Reasoning

Theorem Provers (TPs) prize correctness; but search space for proofs can be huge so many TPs have considered ML.

- CAD work inspired by use of SVMs to select search strategies in the E prover.
- ML used to select the most relevant theorems and definitions when proving a new conjecture in MaLARea.
- Sledgehammer allows for Isabelle/HOL to send goals to a variety of automated TPs and SMT solvers. A relevance filter heuristically ranks the thousands of facts available and selects a subset based on syntactic similarity to the goal.

This work could be the topic of an entire survey talk.

Automated Reasoning

Theorem Provers (TPs) prize correctness; but search space for proofs can be huge so many TPs have considered ML.

- CAD work inspired by use of SVMs to select search strategies in the E prover.
- ML used to select the most relevant theorems and definitions when proving a new conjecture in MaLARea.
- Sledgehammer allows for Isabelle/HOL to send goals to a variety of automated TPs and SMT solvers. A relevance filter heuristically ranks the thousands of facts available and selects a subset based on syntactic similarity to the goal.

This work could be the topic of an entire survey talk. Luckily there will be such a talk this afternoon!

Challenges of using ML in MS

There are challenges in applying machine learning to mathematical software:

- Formulating choices in a way suitable for machine learning. How best to pick from exponentially many choices?
- Obtaining datasets of sufficient size for training. Usually needs thousands of problems for training? Are random problems acceptable?
- Making related choices in tandem: for example the best variable ordering for CAD may change after GB preconditioning! How best to deal with this?

Challenges of using ML in MS

There are challenges in applying machine learning to mathematical software:

- Formulating choices in a way suitable for machine learning. How best to pick from exponentially many choices?
- Obtaining datasets of sufficient size for training. Usually needs thousands of problems for training? Are random problems acceptable?
- Making related choices in tandem: for example the best variable ordering for CAD may change after GB preconditioning! How best to deal with this?

However, there are clearly great potentials also.
Conclusion: Probably we should consider this bandwagon!

The End

For full references see the paper in the ICMS proceedings.

M. England.
Machine Learning for Mathematical Software.
J.H. Davenport, M. Kauers, G. Labahn and J. Urban, eds. Mathematical Software - ICMS 2018, pp. 369-378. (Lecture Notes in Computer Science 10931). Springer, 2018.

Contact Details

`Matthew.England@coventry.ac.uk`

Slides will be available from:
`http://computing.coventry.ac.uk/~mengland/`