Applying machine learning to choose the variable ordering for CAD

Zongyan Huang[1], Matthew England[2], David Wilson[2],
James H. Davenport[2], Lawrence C. Paulson[1] and James Bridge[1]
[1] Computer Laboratory, University of Cambridge
[2] Department of Computer Science, University of Bath

Cylindrical algebraic decomposition (CAD)

- A key tool in computational algebraic geometry
- Widely used in many applications
  - quantifier elimination over the reals
  - robot motion planning
  - programming with complex valued functions
- Introduced by Collins as a alternative to Tarski's decision method for real closed fields (more effective)

A CAD dissects R^n into cells:

- Each described by polynomial relations
- Arranged cylindrically: projection of any two cells into lower coordinates (in the ordering) is equal or disjoint

Variable ordering for CAD

- With CAD we often have a choice as to which variable ordering to use
- The variable ordering is very important: with some problems infeasible with one variable ordering but easy with another (see example suggested by Brown and Davenport)

> A polynomial P_k in $3k+3$ variables such that w.r.t. one variable order there is a CAD of R^{3k+3} for $\{p_k\}$ consisting of 3 cells, while w.r.t. another order any CAD for $\{p_k\}$ has at least 2^{2^k} cells

- Various heuristics are available for picking the ordering, but none is applicable to **all** problems

Sum of Total Degrees (sotd)

- Definition:

> Sum of the total degrees of all monomials in all polynomials in the full projection set

- Suggested by Dolzmann et al., who investigated a variety of basic properties of the polynomials
- Found statistically that sotd was best

Number of Distinct Real Roots (ndrr)

- Suggested by Bradford et al:

> Selects the ordering whose set has the lowest number of distinct real roots of the univariate polynomials in the full projection set

- Assist with examples where sotd failed

Brown

- Labelled after Christopher Brown
- Simple to compute, start with the first and breaking ties with successive ones, eliminate a variable first if:

  1. It has lower overall degree in the input
  2. It has lower total degree of those terms in the input in which it occurs
  3. There is a smaller number of terms in the input which contain the variable

- Very cheap (uses only input data and checks simple properties)

The problem with heuristic choice

- No single heuristic is suitable for all problems
- The best heuristic to use is dependent upon the problem considered
- No obvious relationship between heuristics and problems

Machine Learning

- Deals with the design of programs that can learn rules from data.
- Often a very attractive alternative to manually constructing them when the underlying functional relationship is very complex
- Widely used in many fields:
  - Web searching
  - Face recognition
  - Expert systems

SVM-Light

- In machine learning, supervised learning is the task of inferring a function from labelled data
- Support vector machines (SVMs) are supervised learning models used for classification and regression analysis
- SVM-Light is an implementation of SVMs in C
- Consists of two programs
  - SVM learn: generates a model
  - SVM classify: uses the model to predict the class label and output the margin values

Quantified problems

- 3-variable quantified problems from nlsat dataset
- Output is always a single cell: true or false
- It was not number of cells in the output (but number of cells constructued during the process)

```
Sample QEPCAD input for a quantified problem
(x0,x1,x2)
0
(Ex0)(Ex1)(Ex2) [[((x0 x0) + ((x1 x1) + (x2 x2))) = 1]]
go
go
go
d-stat
go
finish
```

Quantifier free problems

- Quantifier free problems have also been widely used throughout engineering and science
- Separate experiments were run since the results can be quite different (statistics command differs)

Sample QEPCAD input for a quantifier free problem

```
(x0,x1,x2)
3
[[((x0 x0) + ((x1 x1) + (x2 x2))) = 1]]
go
go
d-proj-factors
D-proj-polynomials
go
d-fpc-stat
go
```

Objective and Method

- 7001 problems (3545 problems in training set, 1735 problems in validation set, 1721 problems in test set)
- Heuristics used
  - sotd
  - ndrr
  - Brown
- Machine learning was applied to predict which heuristic will give an "optimal" variable ordering

Data collection

- QEPCAD measured the number of cells generated for each variable ordering
- Each heuristic gives a variable ordering (lexicographical order is applied with multiple choices)
- Determine the heuristic with “optimal” variable ordering

Problem features

- A feature is a measure of the problem that may be expressed numerically
- Each feature vector in the training set was associated with a label +1 (positive example) or -1 (negative example)
- E.g. Brown's heuristic, +1 if Brown's heuristic suggested a variable ordering with the lowest number of cells or -1 otherwise

Identify features

Feature number	Description
1	Number of polynomials
2	Maximum total degree of polynomials
3	Maximum degree of x0 among all polynomials
4	Maximum degree of x1 among all polynomials
5	Maximum degree of x2 among all polynomials
6	Proportion of x0 occurring in polynomials
7	Proportion of x1 occurring in polynomials
8	Proportion of x2 occurring in polynomials
9	Proportion of x0 occurring in monomials
10	Proportion of x1 occurring in monomials
11	Proportion of x2 occurring in monomials

Classification

- Key thing for SVM-Light: select the best model (kernel function) and parameter values
- Base on *Matthews correlation coefficient (MCC)* maximization

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

- Compare the margin values. The classifier with most positive (or least negative) margin was selected.

	condition positive	condition negative
test outcome positive	True positive (TP)	False positive (FP)
test outcome negative	False negative (FN)	True negative (TN)

Criteria: number of problems for which the selected variable ordering is optimal

Table 1: Categorize the problem into a set of mutually exclusive cases by which heuristics were successful

Case	ML	sotd	ndrr	Brown	Unquantified	Quantified
1	Y	Y	Y	Y	399	573
2	Y	Y	Y	N	146	96
3	N	Y	Y	N	39	24
4	Y	Y	N	Y	208	232
5	N	Y	N	Y	35	43
6	Y	N	Y	Y	64	57
7	N	N	Y	Y	7	11
8	Y	Y	N	N	106	66
9	N	Y	N	N	106	75
10	Y	N	Y	N	159	101
11	N	N	Y	N	58	89
12	Y	N	N	Y	230	208
13	N	N	N	Y	164	146

Results

Table2: Proportion of examples where machine learning picks a successful heuristic

sotd	ndrr	Brown	Quantifier free	Quantified
Y	Y	N	79% (>67%)	80% (>67%)
Y	N	Y	86% (>67%)	84% (>67%)
N	Y	Y	90% (>67%)	84% (>67%)
Y	N	N	50% (>33%)	47% (>33%)
N	Y	N	73% (>33%)	53% (>33%)
N	N	Y	58% (>33%)	59% (>33%)

Table 3: Total number of problems for which each heuristic picks the best ordering

	ML	sotd	ndrr	Brown
Quantifier free	1312	1039	872	1107
Quantified	1333	1109	951	1270

Future work

- Use wider data set (more variables, mixed quantifiers)
- Extend the range of features used
- Test more heuristics (e.g. greedy sotd heuristic or combined heuristics) and CAD implementation (e.g. ProjectionCAD, RegularChains in Maple, Mathematica and Redlog)

Are you interested in hearing more about applications of machine learning in computer algebra?

Come to my doctoral talk at 10:30am on Friday !

Thank you!