Minimal Model Complexity Search

Chris McConnell
Carnegie Mellon University

October 5, 1995

The Problem

- Select the model that best describes data
- Select both model form (manual) and fit (automatic)
  - Heuristic data driven: BACON, COPER, FARENHEIT
  - Numeric: polynomials, neural nets
- Same mechanism to learn control knowledge
  - SOAR, Prodigy, Theo, GP
- Tradeoff accuracy and model complexity
  - Solomonoff, SURFER

Minimum Description Length

- Formalization of Occam's razor
- Appoximation to Kolmogorov/Solomonoff Complexity
- $DL(m|d) = L(m) + L(d|m)$
- $P(m|d) = \frac{P(d|m)P(m)}{P(d)}$
- $L(x) = -\log_2 P(x)$
- Relative coding easier than priors
- $L(m) = \Sigma\, c \in mL(c)$
- $L(d|m)$ gaussian model of error
- Best model has the minimum DL
- Orders models by $L(m) \equiv$ control knowledge
- Maximum model length is $L(d)$

Model Vocabulary

Name	*Args*	*Id*	*Inv*	*Commut.*	R_t	P_t	C_t
`X`	Variable				1.0	0.22	2.2
`F`	Parameter				1.0	0.22	2.2
`*`	F X F	1.0	`/`	Y	1.0	0.22	2.2
`/`	F X F	1.0	`*`	N	1.0	0.22	2.2
`LOG`	F		`EXP`	Y	2.0	0.05	4.3
`EXP`	F		`LOG`	Y	2.0	0.05	4.3

- Redundant terms like `(* x x)` are fine

Redundant Models

- Commutative: $x + y = y + x$
- Inverse: $\log e^x = x$
- Unique constant: $x + p_1 - p_2 = x + p_3$
- Signature by random assignment:
  - Constant: $x/x = 1$
  - Distributive: $x * (c_1 + x) = (c_1 + x) + (x * x)$
- Signature after gradient optimized assignment
  - Identity: $x * 1.0 = x$, but $x * p_1 \neq x$
  - $x * p_1 = x/p_2$

Example $x^2 + 3$, $x + p(1)$, $*(2)$

```
Cy    K R S I Ev Co Len Model
00:   0 0 0 0  0  2  98 @X
01:   1 0 0 0 28  2  52 @F
02:   2 0 0 0  0  5  98  (+ X X)
03:   3 0 0 0 12  5  54  (+ X F)
04:   3 1 0 0  0  5      (+ F F) =
05:   4 1 0 0  0  7  54  (* X X)
06:   5 1 0 0 13  7  92  (* X F)
07:   5 2 0 0  0  9      (* F F) =
08:   6 2 0 0  0  9  97  (+ X (+ X X))
09:   7 2 0 0 12  9  64  (+ X (+ X F))
10:   7 2 1 0  0  9      (+ F (+ X X))
      = (+ X (+ X 3.00))
11:   7 3 1 0  0  9      (+ F (+ X F)) =
12:   8 3 1 0  0 10  58  (+ X (* X X))
13:   8 3 1 1 12 10  95  (+ X (* X 3.00))
      = (* X 4.00)
14:   9 3 1 1 11 10  10 @(+ F (* X X))
(+ 3.0143678 (* X X))
```

Test Problems

Name	*In*	*Constants*
	Unary	*Binary*
	Expression	
Example	X	
		+ *
	(+ 3.0 (* X X))	
Boolean	A B C D	
	NOT	AND OR
	(AND (OR A B) (OR C D))	
Regression	X	
	- COS SIN EXP LOG	+ * /
	(+ (* 2.718 (* X X)) (* 3.1416 X))	
Fibonacci	J	0 1 2
	SELF	+ -
	(+ (SELF (- J 2)) (SELF (- J 1)))	
Newton	M1 M2 R	
		+ * /
	(/ (* 6.67 M1 M2) (* R R))	

Example Results

Problem	*Models*			*Rejects*		
	Kept	*Tried*	*Max*	*Gen*	*Sign*	*Inst*
Regression	4519	6121	33224	378	854	370
Regression2	141	194	1584	20	17	16
Boolean	332	926	1520		594	
Boolean2	332	926	1520		594	
Fibonacci	5626	6997	21974	1070	301	
Fibonacci2	5169	6421	73328	316	914	
Newton	4095	8631	62661	323	3166	1047

Problem	*Evals*	*GP Evals*
Regression	36184	306,000
Regression2	1851	306,000
Boolean	926	
Boolean2	926	
Fibonacci	6997	484,500
Fibonacci2	6421	484,500
Newton	133106	

SURFER

Good points:

- Selects the model that best describes data
- Tradeoffs accuracy and model complexity
- Same mechanism could learn control knowledge

Bad points:

- Does not scale well
- Search is not data driven