

## A Little Less AI (in under 500 lines)

**Tim Menzies**

`mailto:youremail@example.com``timm@ieee.org`

`http://yourhomepage.com``http://timm.fyi`

January 5, 2024



# Chapter 1

## 1

For AI and SE, do you know what you are doing? Can we build AI software, succinctly, from just a handful of parts? And can other people understand, critique and improve that system?

Let's check. Let's build an explanation system for a multi-objective semi-supervised optimizer. To say that another way, we want to learn effective human-readable rules for multi-goal problems, after labelling just a handful of examples. These rules have to recognize parts of the problem space where (say) we see the fastest *and* cheapest car designs. And we're going to assume that we can only ask about 20 examples, or less, since if we demand more than that, humans get tired and missables (or they do not have time to answer all our nagging questions).

$N$  things can be Too many choices, not enough time to look at them all.

- e.g. Hundreds of cars in a car yard, you try three, then buy one;
- e.g. You can't test everything – so you just test a few;
- e.g. Software has  $10^9$  of options – but you have time to try a few.

So let's apply *sequential model optimization*:

- $?, ?, ?, ?, ?$
- e.g. Hundreds of cars in a car yard, you try three, then buy one; Some terminology

$$\underbrace{y_1, y_2, \dots}_{\text{dependent variables, goals}} = f(\underbrace{x_1, x_2, x_3, x_4, x_5, x_6, \dots}_{\text{independent variables}})$$

- 
- But dependent variables are more expensive to collect
- e.g. A supermarket has 100 apples. Which ones are tasty? So let's walk data incrementally:

### 1.1 Problem

Too many choices, not enough time to look at them all. line 1.

- adas

---

```

1  function NUM.new(i,at,txt) -- --> NUM; constructor;
2  i.at, i.txt = at or 0, txt or "" -- column position
3  i.n, i.mu, i.m2 = 0, 0, 0
4  i.lo, i.hi = math.huge, -math.huge
5  i.w = i.txt:find"$" and -1 or 1 end
6
7  function NUM.new(i,at,txt) --> NUM; constructor;
8  i.at, i.txt = at or 0, txt or "" -- column position
9  i.n, i.mu, i.m2 = 0, 0, 0
10 i.lo, i.hi = math.huge, -math.huge
11 i.w = i.txt:find"$" and -1 or 1 end
12
13
14
15 function NUM.new(i,at,txt) --> NUM; constructor;
16 i.at, i.txt = at or 0, txt or "" -- column position
17 i.n, i.mu, i.m2 = 0, 0, 0
18 i.lo, i.hi = math.huge, -math.huge
19 i.w = i.txt:find"$" and -1 or 1 end
20
21
22 function NUM.new(i,at,txt) --> NUM; constructor;
23 i.at, i.txt = at or 0, txt or "" -- column position
24 i.n, i.mu, i.m2 = 0, 0, 0
25 i.lo, i.hi = math.huge, -math.huge
26 i.w = i.txt:find"$" and -1 or 1 end
27
28
29 function NUM.new(i,at,txt) --> NUM; constructor;
30 i.at, i.txt = at or 0, txt or "" -- column position
31 i.n, i.mu, i.m2 = 0, 0, 0
32 i.lo, i.hi = math.huge, -math.huge
33 i.w = i.txt:find"$" and -1 or 1 end
34
35
36 function NUM.new(i,at,txt) --> NUM; constructor;
37 i.at, i.txt = at or 0, txt or "" -- column position
38 i.n, i.mu, i.m2 = 0, 0, 0
39 i.lo, i.hi = math.huge, -math.huge
40 i.w = i.txt:find"$" and -1 or 1 end
41
42
43 function NUM.new(i,at,txt) --> NUM; constructor;
44 i.at, i.txt = at or 0, txt or "" -- column position
45 i.n, i.mu, i.m2 = 0, 0, 0
46 i.lo, i.hi = math.huge, -math.huge
47 i.w = i.txt:find"$" and -1 or 1 end
48
49
50 function NUM.new(i,at,txt) --> NUM; constructor;
51 i.at, i.txt = at or 0, txt or "" -- column position
52 i.n, i.mu, i.m2 = 0, 0, 0
53 i.lo, i.hi = math.huge, -math.huge
54 i.w = i.txt:find"$" and -1 or 1 end
55
56
57 function NUM.new(i,at,txt) --> NUM; constructor;
58 i.at, i.txt = at or 0, txt or "" -- column position
59 i.n, i.mu, i.m2 = 0, 0, 0
60 i.lo, i.hi = math.huge, -math.huge
61 i.w = i.txt:find"$" and -1 or 1 end

```

---

## 1.2 Test section one

## 1.3 What is Prolog?

- A programming language associated with artificial intelligence and computational linguistics.
- Based on formal logic.
- Declarative: Describe the problem, not how to solve it.
- Known for its ability to handle symbolic reasoning and knowledge representation.

some text here some text here some text here some text here some text here

asdsa

## 1.4 References

# Bibliography