A Little Less AI (in under 500 lines)

Tim Menzies
mailto:youremail@example.comtimm@ieee.org
http://yourhomepage.comhttp://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 mi stables (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 lets apply *sequential model optimization*:

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

$$\underbrace{y_1, y_2, ...}_{\textit{dependent variables, goals}} = f(\underbrace{x_1, x_2, x_3, x_4, x_5, x_6, ...}_{\textit{indendepentvariables}})$$

-
- But depndent variables are more expesnive to collect
- e.g. A supermarket has 100 apples. Which ones are tasty? So lets walk data incrementally:

1.1 Problem

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

- adas

```
  function NUM.new(i,at,txt) -- --> NUM;  constructor;
  i.at, i.txt = at or 0, txt or "" -- column position
  i.n, i.mu, i.m2 = 0, 0, 0
  i.lo, i.hi = math.huge, -math.huge
  i.w = i.txt:find"-$" and -1 or 1 end

function NUM.new(i,at,txt) --> NUM;  constructor;
  i.at, i.txt = at or 0, txt or "" -- column position
  i.n, i.mu, i.m2 = 0, 0, 0
  i.lo, i.hi = math.huge, -math.huge
  i.w = i.txt:find"-$" and -1 or 1 end



function NUM.new(i,at,txt) --> NUM;  constructor;
  i.at, i.txt = at or 0, txt or "" -- column position
  i.n, i.mu, i.m2 = 0, 0, 0
  i.lo, i.hi = math.huge, -math.huge
  i.w = i.txt:find"-$" and -1 or 1 end


function NUM.new(i,at,txt) --> NUM;  constructor;
  i.at, i.txt = at or 0, txt or "" -- column position
  i.n, i.mu, i.m2 = 0, 0, 0
  i.lo, i.hi = math.huge, -math.huge
  i.w = i.txt:find"-$" and -1 or 1 end


function NUM.new(i,at,txt) --> NUM;  constructor;
  i.at, i.txt = at or 0, txt or "" -- column position
  i.n, i.mu, i.m2 = 0, 0, 0
  i.lo, i.hi = math.huge, -math.huge
  i.w = i.txt:find"-$" and -1 or 1 end


function NUM.new(i,at,txt) --> NUM;  constructor;
  i.at, i.txt = at or 0, txt or "" -- column position
  i.n, i.mu, i.m2 = 0, 0, 0
  i.lo, i.hi = math.huge, -math.huge
  i.w = i.txt:find"-$" and -1 or 1 end

function NUM.new(i,at,txt) --> NUM;  constructor;
  i.at, i.txt = at or 0, txt or "" -- column position
  i.n, i.mu, i.m2 = 0, 0, 0
  i.lo, i.hi = math.huge, -math.huge
  i.w = i.txt:find"-$" and -1 or 1 end

function NUM.new(i,at,txt) --> NUM;  constructor;
  i.at, i.txt = at or 0, txt or "" -- column position
  i.n, i.mu, i.m2 = 0, 0, 0
  i.lo, i.hi = math.huge, -math.huge
  i.w = i.txt:find"-$" and -1 or 1 end

function NUM.new(i,at,txt) --> NUM;  constructor;
  i.at, i.txt = at or 0, txt or "" -- column position
  i.n, i.mu, i.m2 = 0, 0, 0
  i.lo, i.hi = math.huge, -math.huge
  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