dtControl 2.0

Explainable Strategy Representation via Decision Tree Learning Steered by Experts

Pranav Ashok, Mathias Jackermeier, Jan Kretinsky, Christoph Weinhuber, **Maximilian Weininger** and Mayank Yadav

Talk in one slide

Small

Explainable

https://breezecomputer.com/wp-content/uploads/2018/12/1.-Huele-GameCube-Controller-2-Pack.jpg

Talk in one slide

dtControl 2.0

Small

Explainable

https://breezecomputer.com/wp-content/uploads/2018/12/1.-Huele-GameCube-Controller-2-Pack.jpg

Talk in one slide

Why?

What?

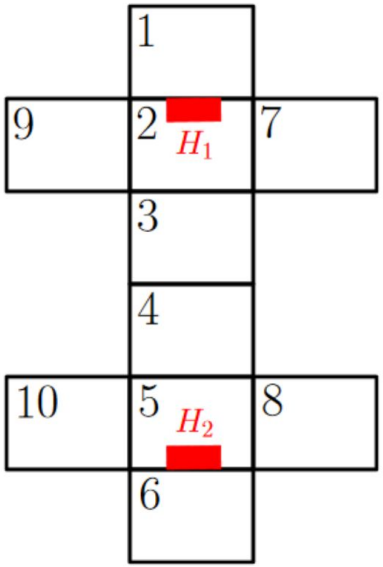
dtControl 2.0

How?

- Small
- Explainable

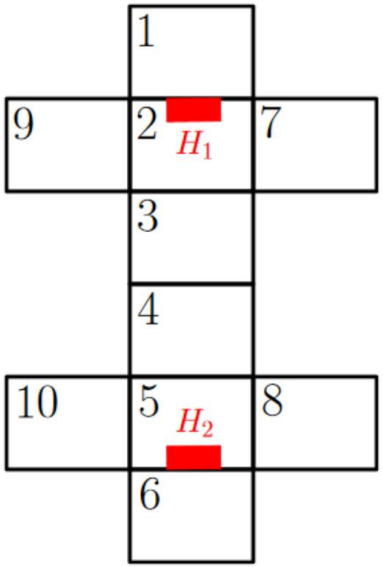
https://breezecomputer.com/wp-content/uploads/2018/12/1.-Huele-GameCube-Controller-2-Pack.jpg

Controller by example



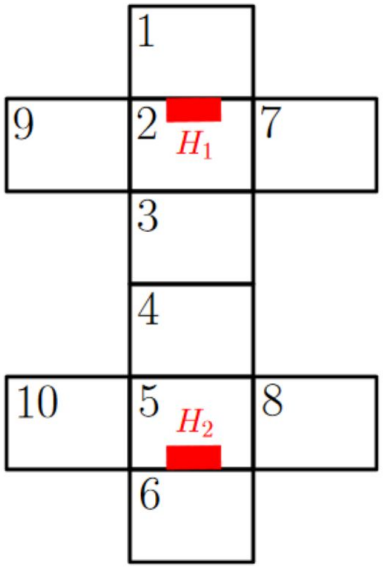
$C : S \rightarrow A$

Controller by example



$C : S \rightarrow A$

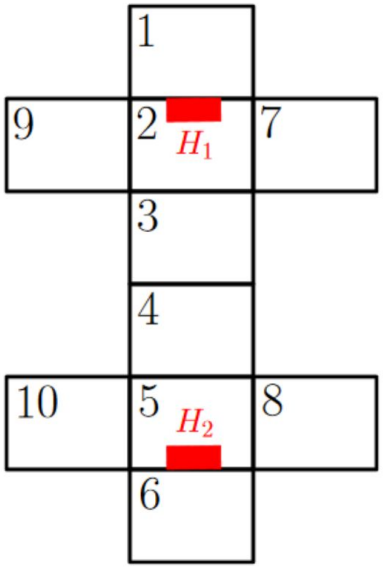
Controller by example



$C : S \to A$

```
#SCOTS:v0.2
...
#BEGIN:59049 9
84 8
85 8
86 8
87 6 7 8
88 6 7 8
89 6 7 8
93 8
94 8
95 7 8
96 6 7 8
97 6 7 8
98 6 7 8
102 8
103 7 8
104 7 8
105 6 7 8
106 6 7 8
107 6 7 8
111 8
112 8
```

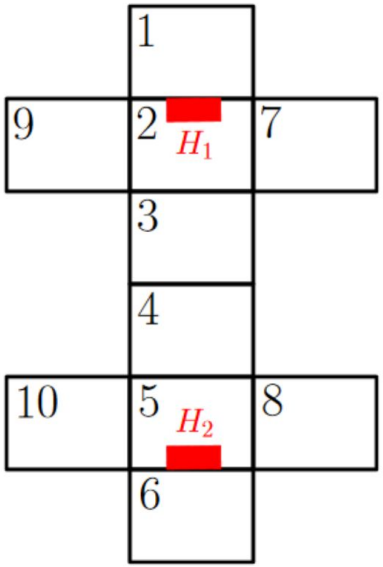
Controller by example



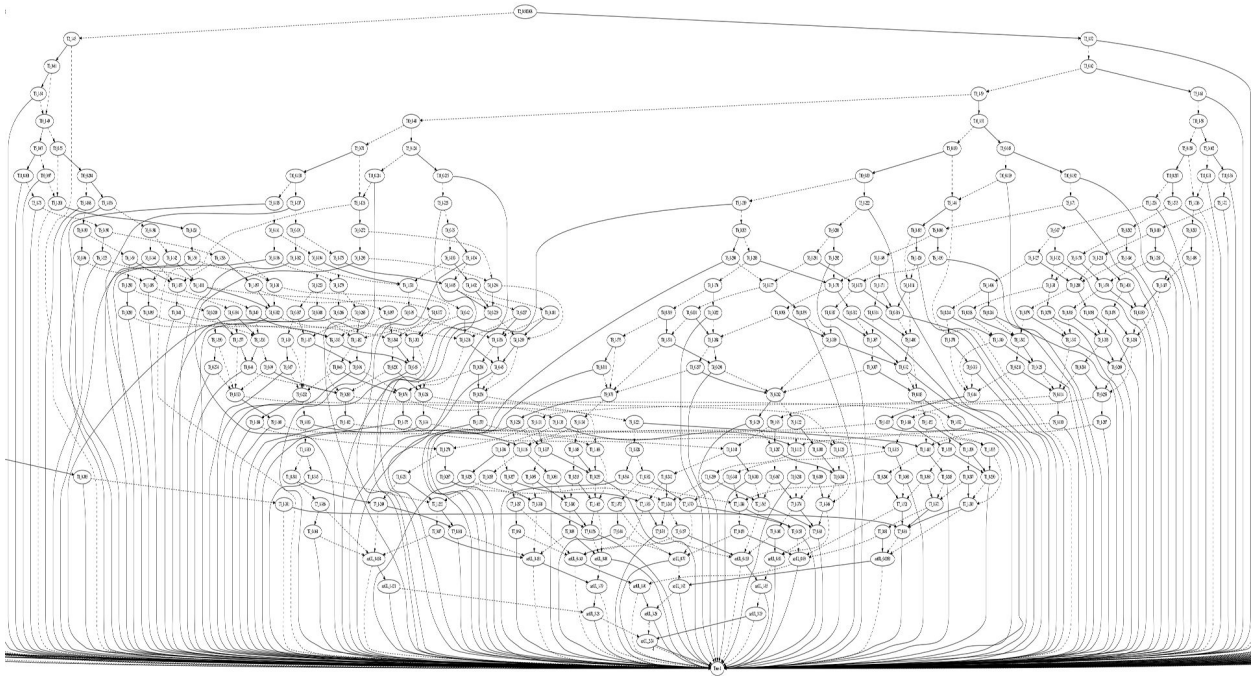
$C : S \to A$

```
#SCOTS:v0.2
...
#BEGIN:59049 9
84 8
85 8
86 8
87 6 7 8
88 6 7 8
89 6 7 8
93 8
94 8
95 7 8
96 6 7 8
97 6 7 8
98 6 7 8
102 8
103 7 8
104 7 8
105 6 7 8
106 6 7 8
107 6 7 8
111 8
112 8
```

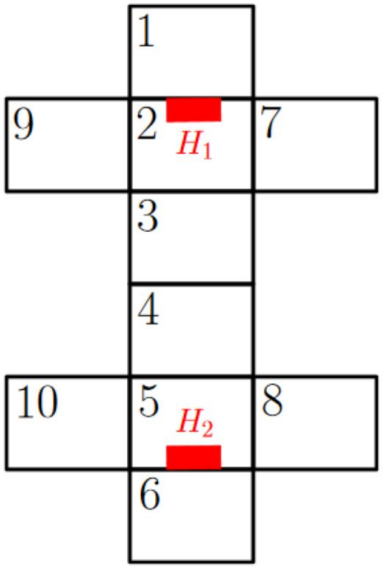
Controller by example



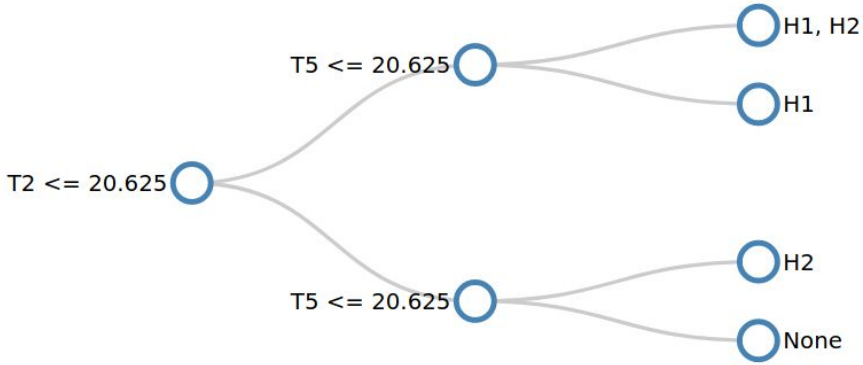
$C : S \to A$



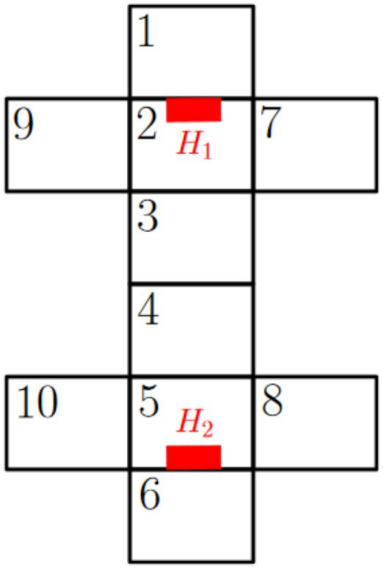
Controller by example



$C : S \rightarrow A$



Controller by example

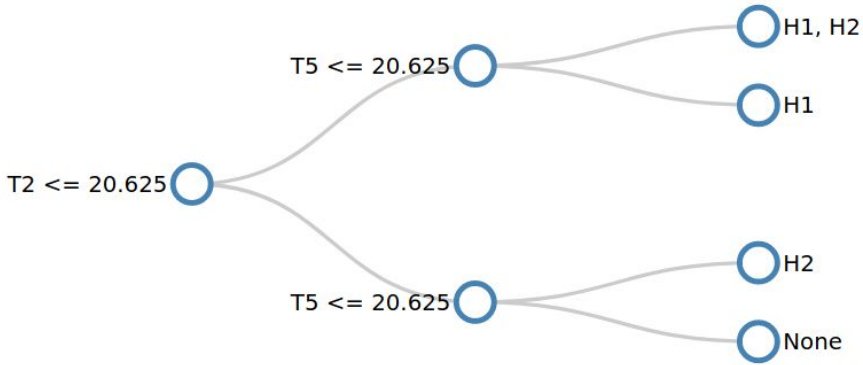


$C : S \rightarrow A$

Memory efficient

Small

Executable



Explainable

Helps debugging & modeling

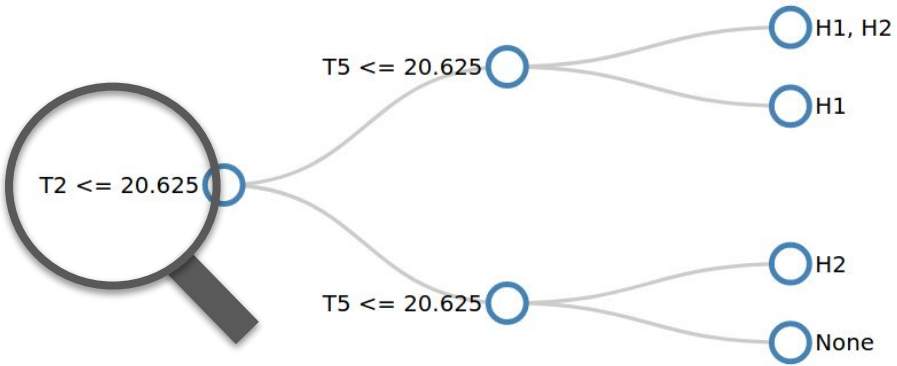
Lowers deployment cost

How?

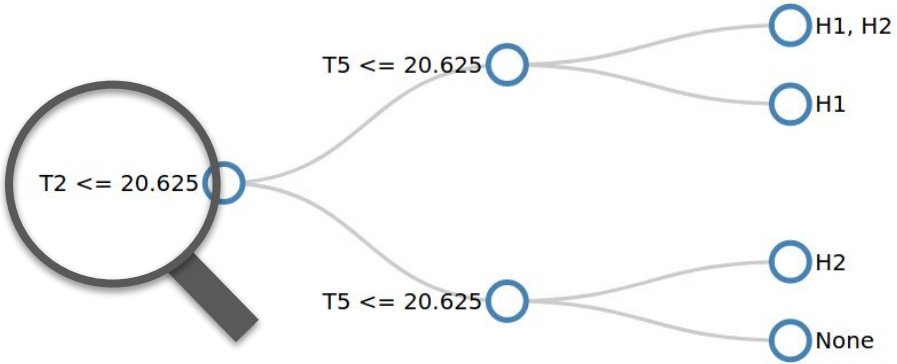
How?

dtControl 2.0

Inner workings

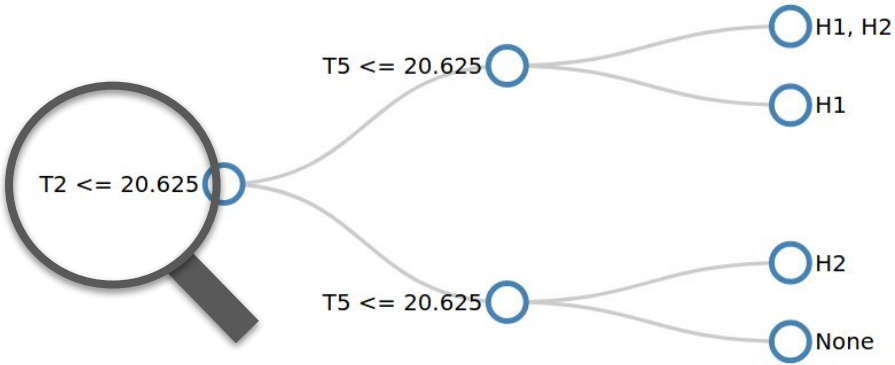


Inner workings



HSCC video:
https://www.youtube.com/watch?v=K6d3pS6Ege0
(slide 23-31)

Inner workings

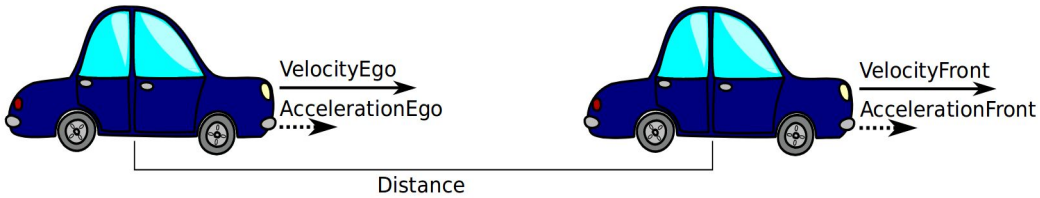


Axis-aligned
T2 <= 20.625

Linear
0.5*T2 + 1.3*T5 <= 20.625

Algebraic
T2^4 * log(T5) - sqrt(T7) <= 20.625

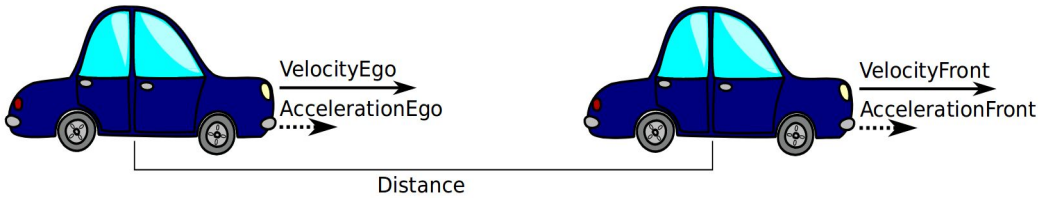
Algebraic predicates



Safe controller

Lookup Table	295,695	rows
BDD	2,400	nodes
dtControl 1.0	374	nodes

Algebraic predicates

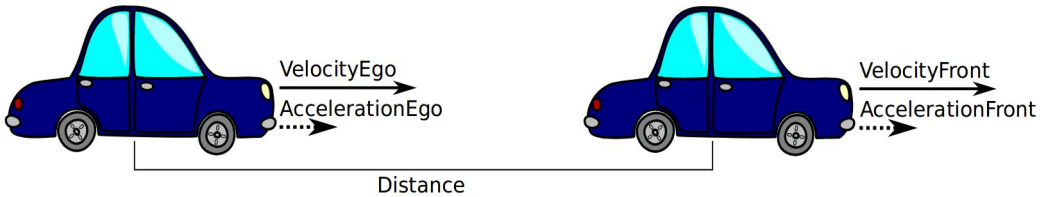


Domain knowledge

$d = \frac{1}{2} at^2 + ut + d_0$

$v = u + at$

Algebraic predicates



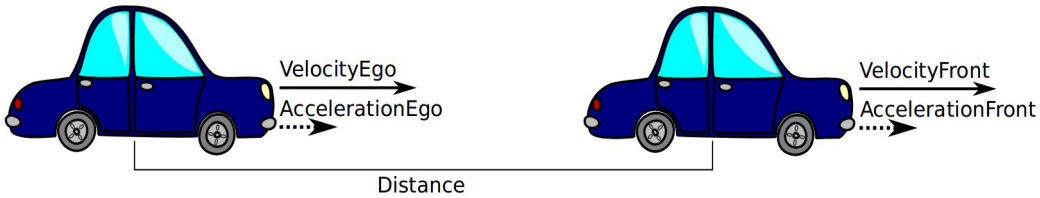
$$d - 2 + \left(\frac{-4 - v_e - 2t_f)}{-2}\right)^2 + (-6 - v_e)\left(\frac{-6 - v_e - 2t_f)}{-2}\right) \leq 5$$

Domain knowledge

$d = \frac{1}{2} at^2 + ut + d_0$

$v = u + at$

Algebraic predicates



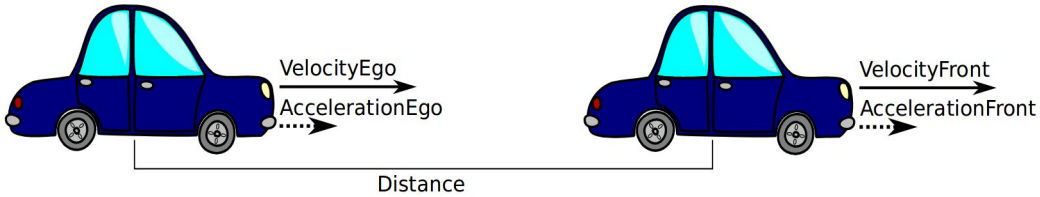
Worst-case distance (accelerate) ≤ 5

Domain knowledge

$d = \frac{1}{2} at^2 + ut + d_0$

$v = u + at$

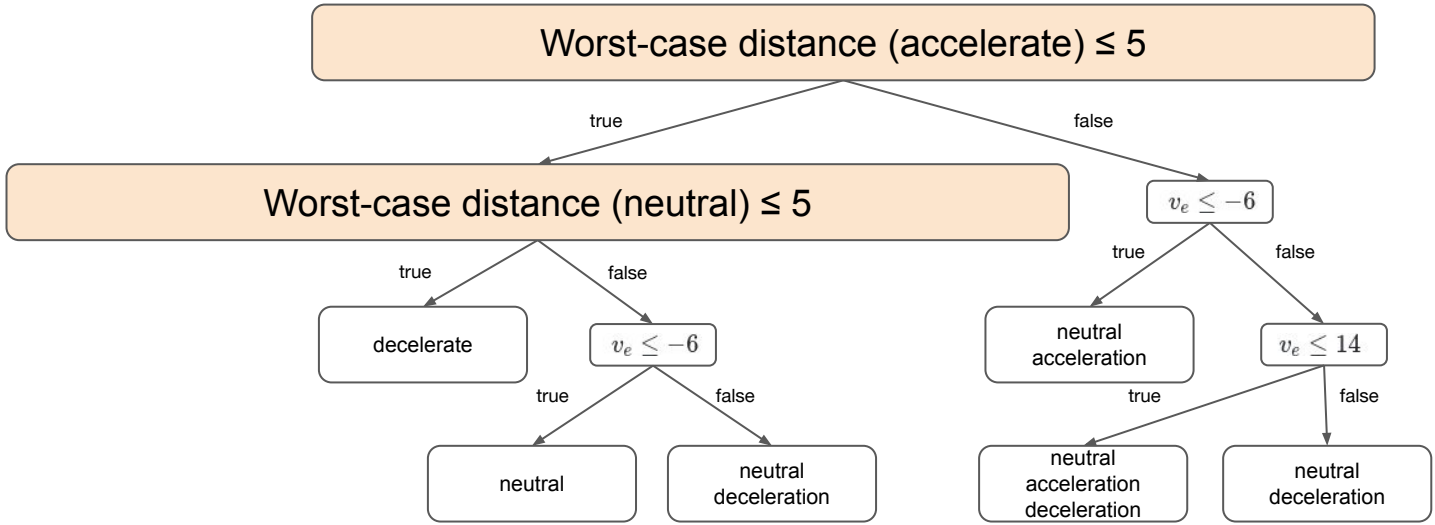
Algebraic predicates



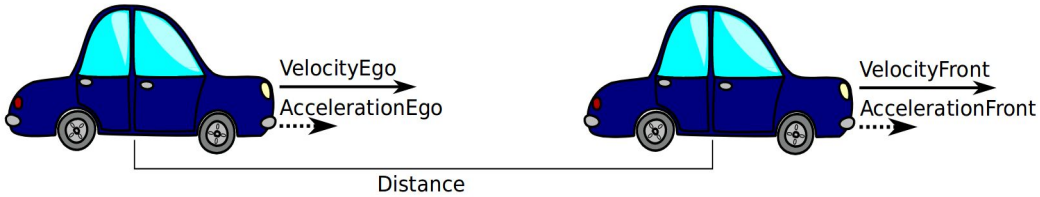
Domain knowledge

$d = \frac{1}{2} at^2 + ut + d_0$

$v = u + at$



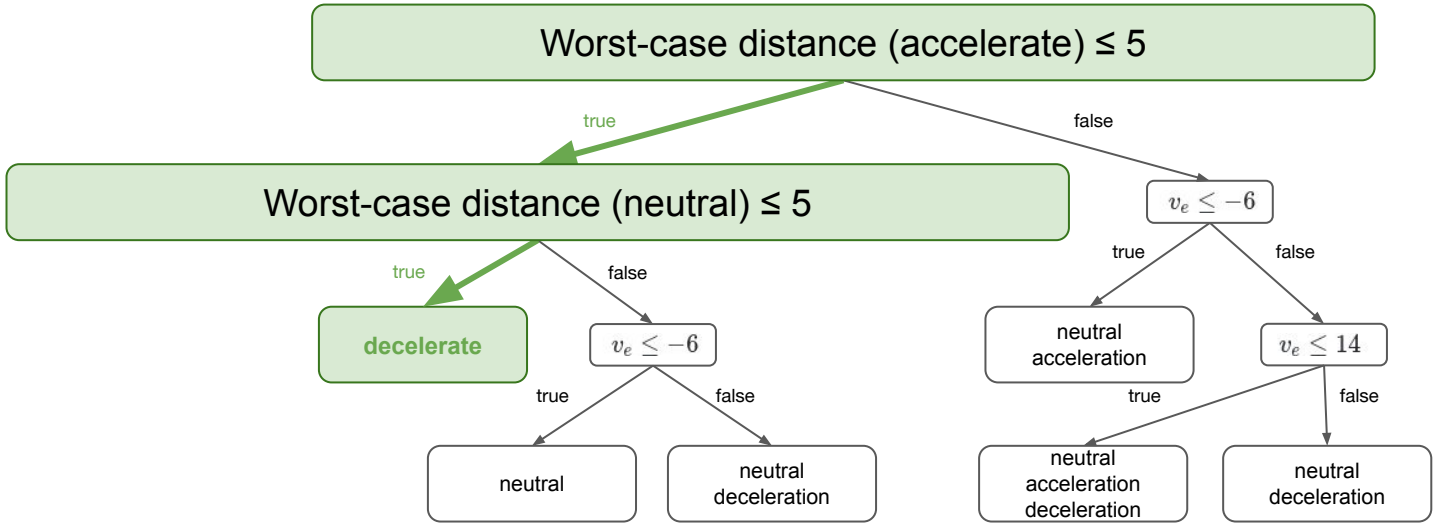
Algebraic predicates



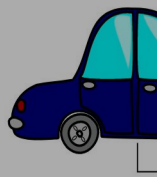
Domain knowledge

$d = \frac{1}{2} at^2 + ut + d_0$

$v = u + at$



Alge



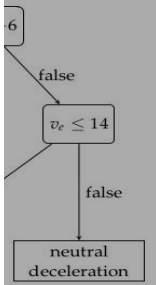
Domai
d = ½ a
v = u +

Distilling a Neural Network Into a Soft Decision Tree

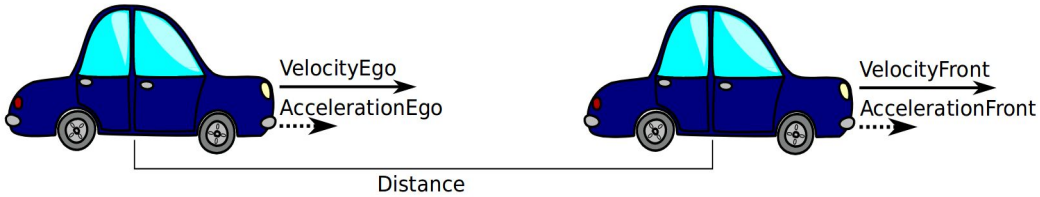
Nicholas Frosst, Geoffrey Hinton

Google Brain Team

Abstract. Deep neural networks have proved to be a very effective way to perform classification tasks. They excel when the input data is high dimensional, the relationship between the input and the output is complicated, and the number of labeled training examples is large [Szegedy et al., 2015, Wu et al., 2016, Jozefowicz et al., 2016, Graves et al., 2013]. But it is hard to explain why a learned network makes a particular classification decision on a particular test case. This is due to their reliance on distributed hierarchical representations. If we could take the knowledge acquired by the neural net and express the



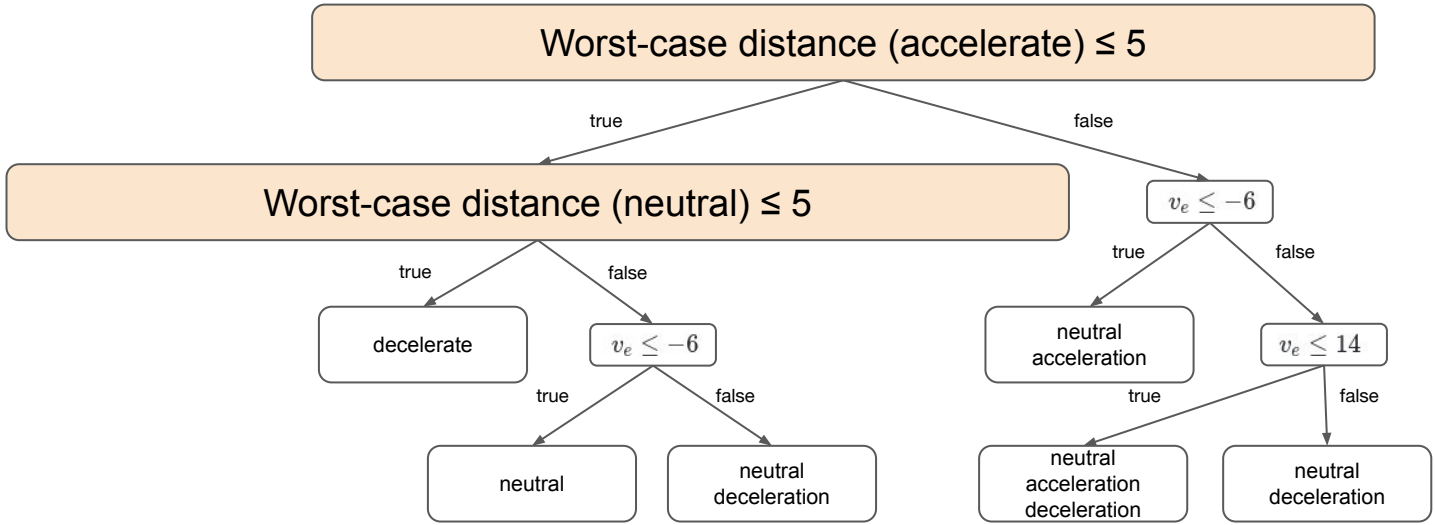
Algebraic predicates



Domain knowledge

$d = \frac{1}{2} at^2 + ut + d_0$

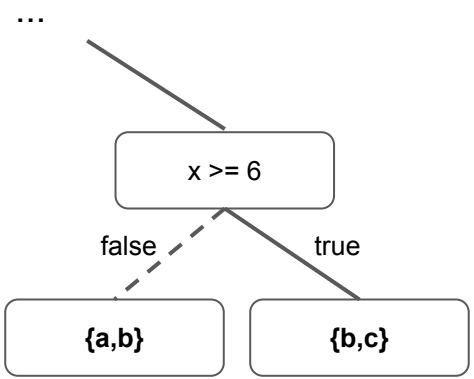
$v = u + at$



Demo

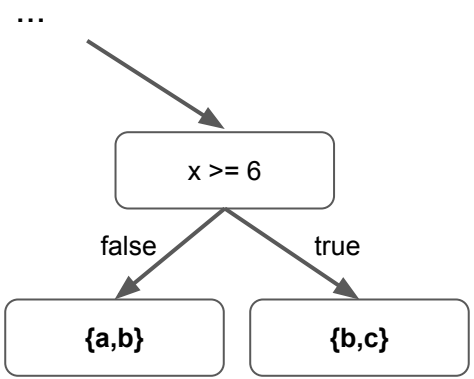
- This is how the tree looks in our awesome new GUI
- We can scroll, collapse etc. to explore the tree
- This is how we made the tree:
  Choose controller file - Note we support Storm, PRISM, Scots, Uppaal and csv
  Choose preset (lots of automatic, we now want user-defined)
  Enter your cool predicates (first an axis-aligned, then the rest with a good mix of explanation and Ctrl+V)
  If this was scary: Documentation
- Add, play; while playing, explain benchmarking
- View -> Result we showed earlier.
- Ok, how did we get these fancy predicates? 1. Domain knowledge, understanding 2. Trial and error. Show interactive tree building.
- Much more to help us understand controller and DT, like running simulations and more ways to affect predicate construction or changing trees on the fly. -> Delegate to Docs, papers and QEST video

Determinization

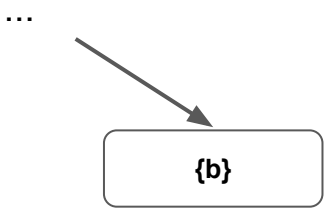


Permissive

Determinization



Permissive



Determinized

Experimental results

Case study	States	BDD	dtControl 1.0	dtControl 2.0
cartpole	271	127	11	**7**
10rooms	26,244	128	**7**	**7**
helicopter	280,539	870	221	**123**
cruise-latest	295,615	1,448	**3**	**3**
dcdc	593,089	381	9	**5**
truck_trailer	1,386,211	**18,186**	42,561	31,499
traffic_30m	16,639,662	TO	127	**97**

Conclusion

```
#SCOTS:v0.2
#TYPE:STATICCONTROLLER
#SCOTS:STATE_SPACE
#TYPE:UNIFORMGRID
#MEMBER:DIM
10
#VECTOR:ETA
#BEGIN:10
1.25
1.25
1.25
1.25
1.25
1.25
1.25
1.25
1.25
1.25
```

dtcontrol.model.in.tum.de

dtControl 2.0

+ Algebraic predicates
+ GUI
+ Better determinization

Small

Explainable

https://breezecomputer.com/wp-content/uploads/2018/12/1.-Huele-GameCube-Controller-2-Pack.jpg

Conclusion

```
#SCOTS:v0.2
#TYPE:STATICCONTROLLER
#SCOTS:STATE_SPACE
#TYPE:UNIFORMGRID
#MEMBER:DIM
10
#VECTOR:ETA
#BEGIN:10
1.25
1.25
1.25
1.25
1.25
1.25
1.25
1.25
1.25
1.25
```

dtcontrol.model.in.tum.de

dtControl 2.0

+ Algebraic predicates
+ GUI
+ Better determinization

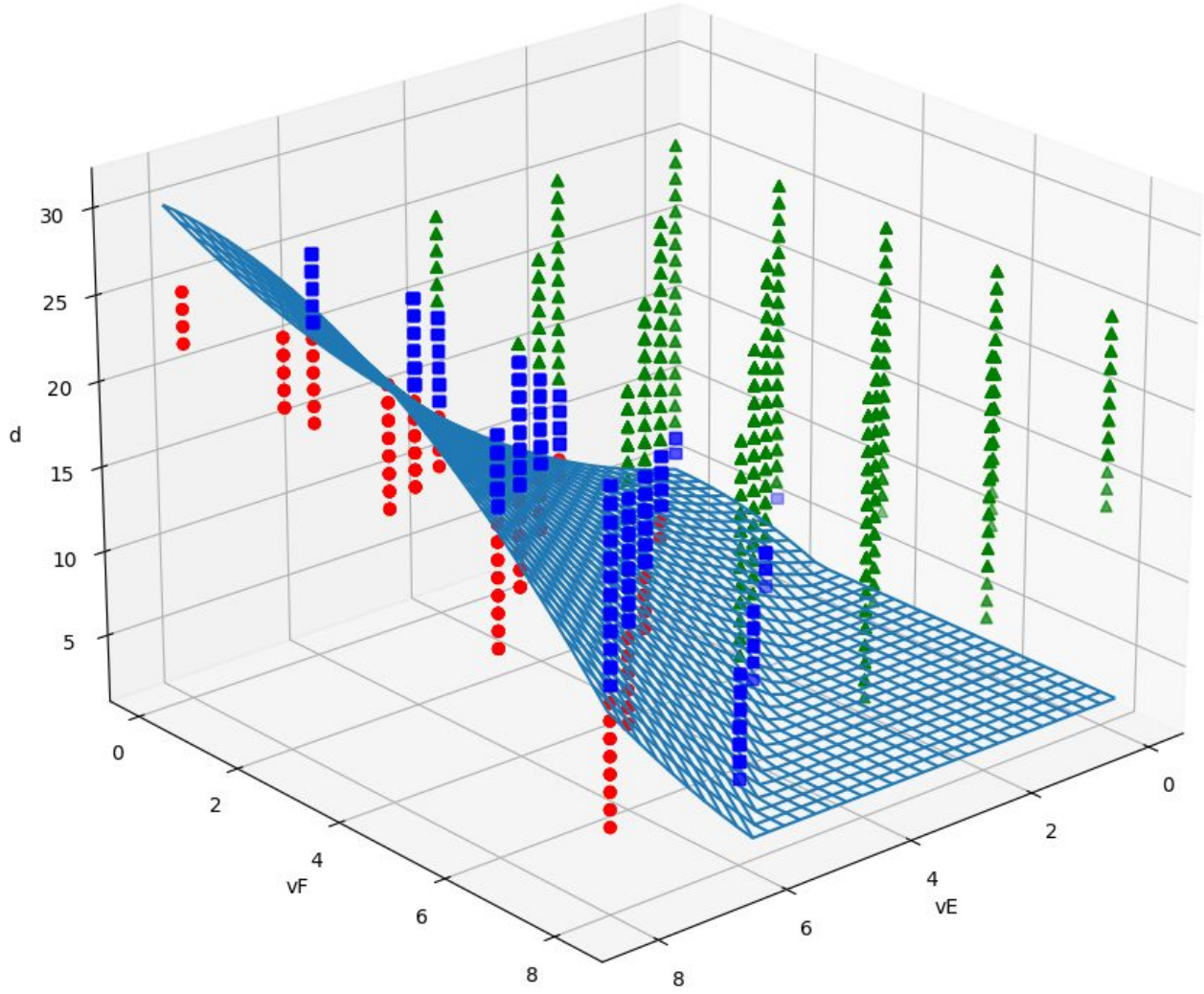
Small

Explainable

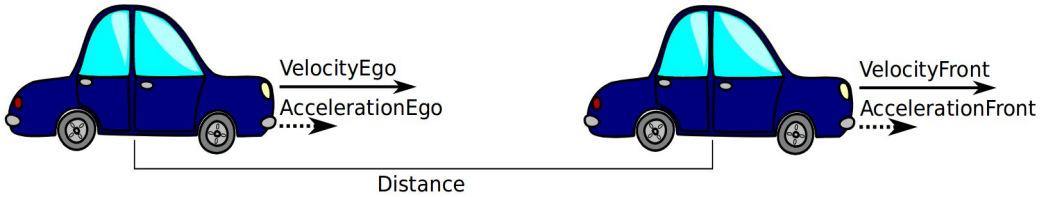
Thank you!

https://breezecomputer.com/wp-content/uploads/2018/12/1.-Huele-GameCube-Controller-2-Pack.jpg

More information


30
25
20
d
15
10
5
0
2
4
vF
6
8
0
2
4
vE
6
8

Algebraic predicates



Domain knowledge

$d = ½\ at^2 + vt + d_0$

$v = at$