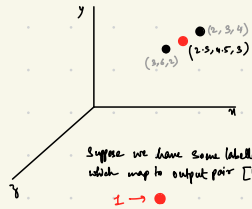
Adaptive Range Based Classification Continued.



Suppose we have some labelled inputs which map to output pair [1,2]

1 → ●

2 → ●

model is dumb initially,

Now consider some points.

(2.5, 4.5, 3) → 1

(3, 6, 2), (2, 3, 4) → 2

On first run, model understands and maps the first point with its result.

Second run also seems to be working smoothly as there are no conflicts.*

	1st	2nd	
Initial range of x →	3	[2, 3]	Both map to 2
y →	6	[3, 6]	
z →	2	[2, 4]	
	lower	upper	

conflict: I mean a position which is supposed to be mapped to a result, but is mapped to some other output.

Now comes a conflict,

from what we see the pt. [2.5, 4.5, 3] must map to '2' also, but the result we see is '1' so we need to resolve it.

So what we do,

Locate the ranges (can be modified with more functions)

$$\begin{matrix} x \\ y \\ z \end{matrix}\begin{bmatrix} 2,3 \\ 3,6 \\ 2,4 \end{bmatrix}\begin{matrix} x \\ y \\ z \end{matrix}\begin{bmatrix} 2.5 \\ 4.5 \\ 3 \end{bmatrix}$$

Like we split equally but ranges can be split in a different ratio with some heuristic funct.

get a new range

$$\begin{bmatrix} 2 \to 2.25 \\ 3 \to 3.75 \\ 2 \to 2.5 \end{bmatrix}$$ maps to 2

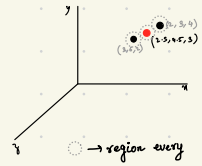
$$\begin{bmatrix} 2.25 \to 2.5 \\ 3.75 \to 4.5 \\ 2.5 \to 3 \end{bmatrix}$$ maps to 1

$$\begin{bmatrix} 2.5 \to 2.75 \\ 4.5 \to 5.25 \\ 3 \to 3.5 \end{bmatrix}$$ maps to 1

$$\begin{bmatrix} 2.75 \to 3 \\ 5.25 \to 6 \\ 3.5 \to 4 \end{bmatrix}$$ maps to 2

And Basically this is how model will learn from data and draw classifications.

This research is a continued version to my previous ML model in C which also works on the same principle.

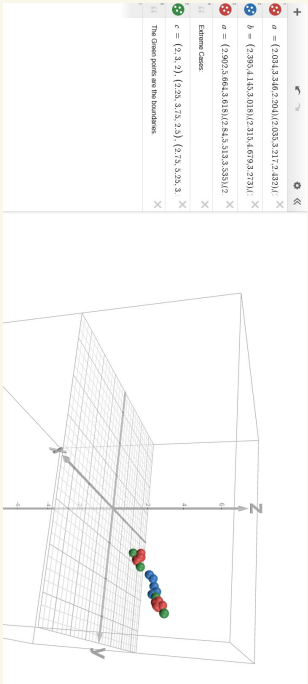


→ region every single point covers.

* all points will have some range, i.e covers some region.

After training this model it will distribute the 3d space (in this case) in small regions adapting to even the slightest of changes, will not be overfitting itself as the inputs will be classified based on their nearest neighbour's ranges.

Sample after training:



Let's talks about another important case, i.e, what if we have two points and a 3rd point lies in a position which is equally distant from 'A' and 'B'. What color will it take?

Well, it has a 50% chance so we can assign random and if we are wrong we can always adjust.

Another case can be what if (2, 3, 4) → 'A' is passed and on another input (2, 3, 4) → 'B' is passed, Well this will update old known data (2, 3, 4) → 'A' with 'B'.

* → This research takes some ideas of k-nearest neighbours but does not follow it blindly. This introduces new form of classification with capability to adapt in higher dimensions and provide better accuracy.

Cons:

— Needs huge amounts of data for increased perfection!

→ Debasis (2024)