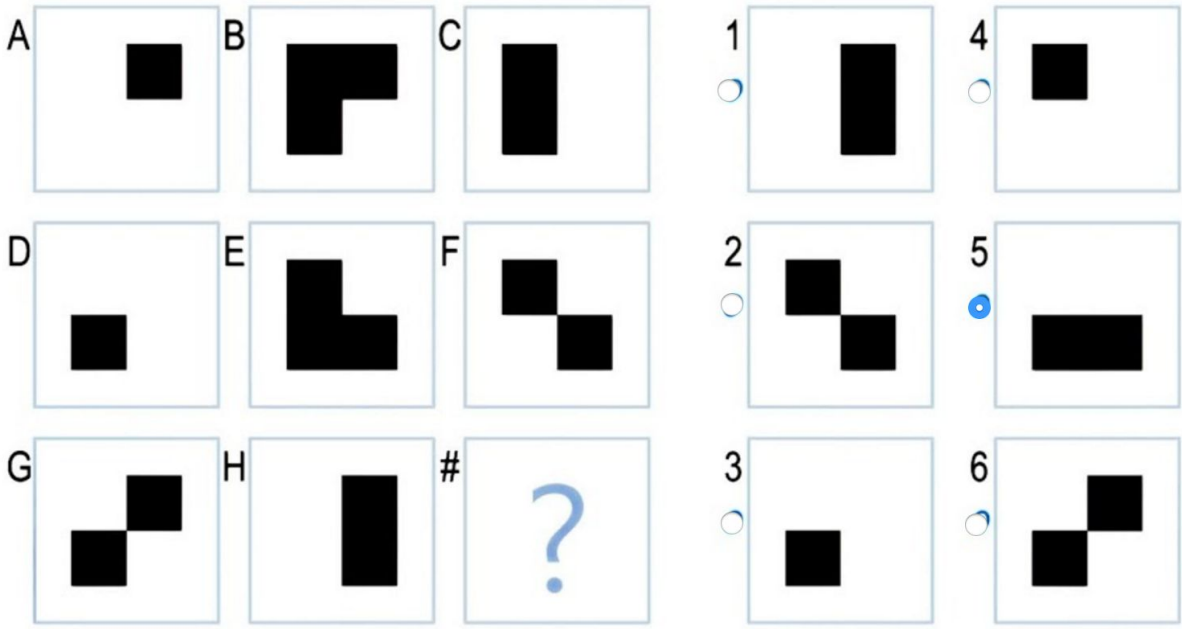
Assignment 2: Learning by recording cases with a blend of incremental concept learning

Introduction:

This project is to create an Artificially-Intelligent agent to solve Raven's Progressive Matrices. Raven's Progressive Matrices (RPMs) are a commonly-used tool to measure intelligence. Consistency heuristic is a generally used technique in learning by recording cases. According to this heuristic, solution can be obtained by finding the most similar cases from a given set of cases. Using incremental concept learning, we can abstract a concept out of these recorded examples and use the concept in solving a new problem.

Problem:
We'll pick one of the toughest problems in Raven's progressive matrices.



Here, the problem is to find the last shape in (G, H, ?) set given (A, B, C) and (D, E, F) sets. There is no clear symmetry in the figures from given sets that can be applied directly. The relationship between figures is very logical. A and B have the squares that are XOR'ed to obtain C. The filling present in same area for A and B wouldn't appear in C. Rest all filling is carried from A or B to C. Same is the case with D, E and F. It's difficult for an agent to guess this XOR operation. We'll see how an agent can be designed that would be able to solve this problem from the learning in two cases (A, B, C) and (D, E, F).

Solution:

Let the black filling be considered to spread within a square area with four quadrants Q1, Q2, Q3, Q4 corresponding to top-left, top-right, bottom-left and bottom-right respectively. Our agent first finds the bit representations of each figure and derives a common feature from bit

representations for each case. Since there are four quadrants, each figure will be a 4-bit representation. If a particular shape has a filling/darkness in some quadrant, that particular quadrant takes a bit ‘1’, and ‘0’ otherwise. This filling check can be done by image processing. Agent thresholds to find dark areas of appropriate quadrant size. Image processing techniques like color detection of object[1] or blob detection[2] can be used to find color of quadrants or quadrants covered by filling respectively.

Agent then adds up the number of ‘1’ bits in corresponding positions for 4 bits in three bit-string representations of both cases individually and stores the counts. There’ll be a total of 4 bit-counts for case 1 and 4 more for case 2, a total of 8 non-negative integers. Common feature that’s deduced from these eight integers is that the bit count is ‘0’ or ‘2’. It applies the same mechanism on answer options until the bit count is only ‘0’ or ‘2’ to finalize the correct option.

Execution:

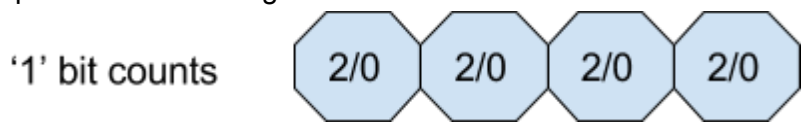
A	0	1	0	0
B	1	1	1	0
C	1	0	1	0
Sum-A,B,C	2	2	2	0
D	0	0	1	0
E	1	0	1	1
F	1	0	0	1
Sum-D,E,F	2	0	2	2
G	0	1	1	0
H	0	1	0	1
# - Option 1	0	1	0	1
Sum-G,H,1	0	3	1	2
# - Option 2	1	0	0	1
Sum - G, H,2	1	2	1	2
# - Option 3	0	0	1	0
Sum - G, H,3	0	2	2	1
# - Option 4	1	0	0	0
Sum - G, H,4	1	2	1	1

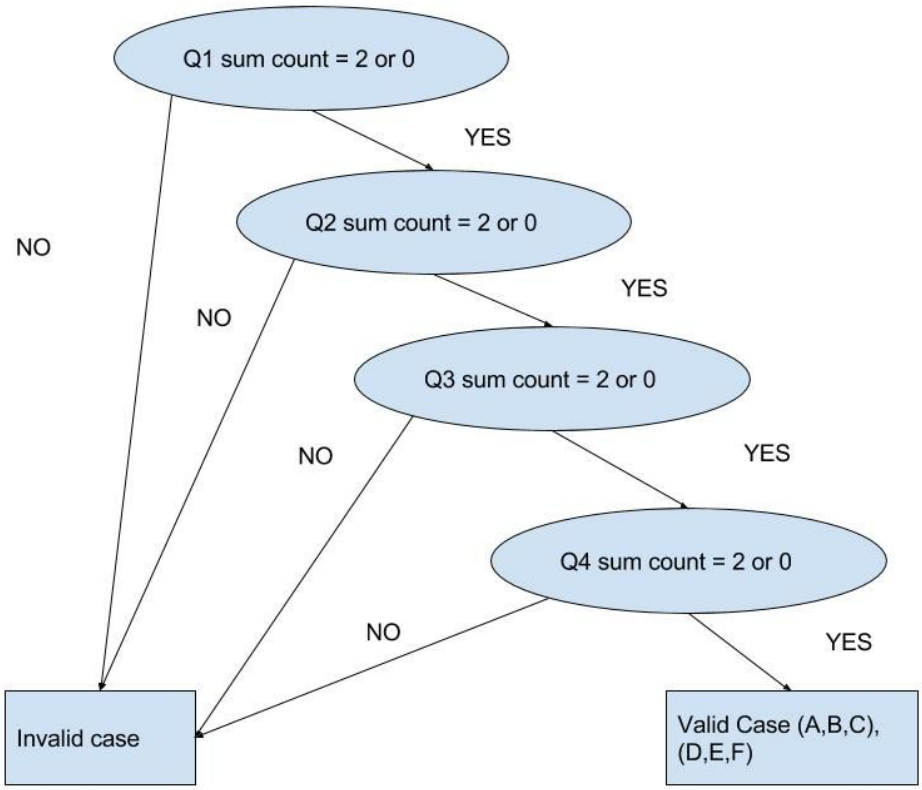
# - Option 5	0	0	1	1
Sum - G, H,5	0	2	2	2
# - Option 6	0	1	1	0
Sum - G, H,6	0	3	2	1

We can see from above table created by agent that the reddened sums are the failed options since the bit counts include integers other than 0 and 2. Green sum for option 5 is the correct option, in sync with case 1 and case 2 learning.

Agent adapts the two cases information to abstract a concept that bit counts can be only ‘2’ or ‘0’.

Concept on bit count string and decision tree for the same are as below:





All options are run through the above bit string creation, finding ‘1’ bit counts string and decision tree traversal or the concept check to evaluate if it’s a valid option or not. Agent finds option 5 as correct answer for the current problem.

Conclusion:
The blend of different techniques is helpful in solving tricky problems like this. We mixed the principles of learning by recording cases in recording the bit strings, case reasoning by adaptation to ‘1’ bit count strings and incremental concept learning on sum bit strings.

References:
[1] https://www.robotix.in/tutorial/imageprocessing/colour/
[2] https://www.robotix.in/tutorial/imageprocessing/blob/