CS 7637 Project 2 Reflection:

Abhijeet Gaurav
agaurav@gatech.edu

Introduction

The goal of KBAI course is to learn to build human like intelligence. In this journal we'll discuss my approach to building an AI agent that will solve Raven's Progressive Matrices (RPM) which were used in the past to test abstract reasoning and human intelligence. This journal covers the entire design and implementation of my AI agent that will solve the 3x3 RPM problems and will build up (to some extent) on my AI agent that solved 2x2 RPM problems for Project 1. 3x3 RPM problems are similar to 2x2 RPM problems where there is some pattern/relationships among a set of given 8 figures (fig A to fig H). Using the same observed relationship between these 8 figures, the AI agent has to identify the 9th figure that best follows a similar relationship. This optimal figure that completes the relationship is given as one of the 8 options (fig 1 to fig 8) to the AI agent to pick from.

Building Project 2 to solve on top of Project 1:

My agent will only use visual representation as an input to my AI Agent. I initially decided to scale my 2x2 RPM problem solver to solve 3x3 RPM problems as I thought my 2x2 RPM problem solver was very robust (although it overfitted to some extent on the Set B basic problems. My 2x2 AI agent use to do transformations on A, compare it with B/C to see if there is a horizontal/vertical relationship. If horizontal/vertical relationship was found, same transformation on B/C was applied and then compared to 6 options. Similarly, it also used to do transformations on B, compare it with C to determine if there is a diagonal relationship. If diagonal relationship was found, similar transformation was done to A and transformed A was compared with all the options. Each option was assigned a score. Option with the highest score was my agent's answer. Applied transformations were either trivial (rotation/identical/reflection) or non-trivial (compare difference of difference image/fill image with color transformation). Using this approach my AI agent solved 12/12 basic 2x2 problems, 9/12 test 2x2 problems, 8/12 raven 2x2 problems; 7/12 2x2 challenge problems. My original idea was to divide 3x3 problems in chunks of 2x2 problems and solve each of the sub 2x2 problems to achieve a solution for 3x3 problems (which to my surprise didn't work at all because 3x3 RPM problems give to us in Project 2 were quite different from 2x2 RPM problems given to us in Project 1). Therefore, I had to start from scratch to some extent for project 2 by taking a few concepts that I implemented for my 2x2 solver. For my new 3x3 RPM problem solver, I used scoring mechanism based on dark pixel density (to some extent taken from my 2x2 RPM problem solver where I used white to dark pixel ratio instead) to determine how dark pixels increased/decreased in horizontal/vertical 3x3 relationships and scored options based on this metric, diff image scorer (taken from my 2x2 problem solver but the underlying logic is a bit different) which subtracts one image from another which helps in identifying shape change in horizontal/vertical relationships, reflection scorer (similar to my 2x2 problem solver but the underlying logic is a bit different) which checks if the figures follow any reflection relationship, slice image and compare scoring mechanism which slices the image horizontally/vertically into two parts and then left slice of each image is compared to right slice (and vice-versa) of the other images to determine if they are equal. Each of these scoring mechanisms will be explored in detail in my submission sections of this Journal.

First Submission

The first submission was sent on 2019-10-26 20:08:16 UTC. My first approach to solving the 3x3 problems , was to divide the problems into 4 smaller sets of 2x2 RPM problems and solve them independently using my 2x2 RPM solver. The idea was to consider 2x2 relationship between these smaller 2x2 problems from the whole 3x3 problem: E-F-H-?; B-C-H-?; D-F-G-?; A-C-G-? . Each of these sub 2x2 problems would score the options independently from one another. Each of the scores from these 2x2 problems would be given the same weight and would be added to form a cumulative score. The option with the highest score would be the Agent's final answer. My 2x2 RPM first tries to build possible relationships horizontally, vertically and diagonally based on Generate and Test method. The generator generates possible transformation on A(to check horizontal/vertical relationship) and B (to check diagonal relationship) by rotation, reflection, diff image transformation (The goal of diff image scorer is to compare subtracted images. Basically, image (A-B) is compared with image (C-option) in horizontal relationship and (A-C) is compared with image (B-option) in vertical relationship and similarly for diagonal relationship), color filing transformation (by filling the lower white to black pixel ratio figures with black color) and a smart tester which tests if the generated figure is similar to one of the other figures (A and/or B and/or C). If the tester marks the transformed figure (say A), similar to the other given figure (say B), a similar transformation is applied to the other remaining figure (say C) and compared with all the options. The tester then checks each option if it matches the transformed figure (transformed C in our case). The option that matches the most gets the highest score.

Agent comparison to humans approaching the problem: I think my agent does not approach 3x3 problem as a human overall. I don't divide 3x3 problem in sets of 2x2 problem and solve them individually. However, if you just look at individual 2x2 problem separately, my AI agent follows the exact same steps that I follow to solve a 2x2 RPM problem. It tries to rotate/reflect and then compare. It gives more weightage to the reflection option as opposed to the rotation problem. I do the same thing, in my mind, I am rotating and reflecting the figure, before I come to a solution.

Performance:

Problem Set C: Basic problems: 2/12 ; Test problems: 2/12; Ravens problems: 3/12; Challenge problems: 3/12. Problem Set B: Basic problems: 12/12 ; Test problems: 9/12; Ravens problems: 8/12; Challenge problems: 7/12. My agent performed miserably on Problem Set C using this approach of dividing 3x3 RPM problem into sub 2x2 problem. The AI agent continued to do well on Problem Set B as there hasn't been any changes in the underlying logic (that I implemented in Project 1). The approach of dividing 3x3 RPM problem, failed because when we create sub 2x2 problems, the relationship between all the figures of the entire 3x3 RPM problem is not preserved. Also, I saw some race condition during finalizing the final answer option as one subset of 2x2 problem choose one answer while the other subset chose another answer. The other reason of this implementation not working well was because the 3x3 problems are quite different than 2x2 problem. 3x3 RPM problems given in the Project2 do not have rotation (and a direct reflection) relationship. Also, there are no 3x3 problems of filling the entire figure with black color. I came to know about these issues while I ran this 3x3 RPM solver locally on Set C: Basic problems, I still wanted to test my AI agent on Test and Raven's problem to make sure that these problem sets do not have problems based on rotation (and a direct reflection), filling figure problems. Overall, the agent only solved, Basic Problem C-01 (which has identical figure horizontal relationship) and Basic Problem C-07 (which has reflection property). In terms of **efficiency**, my agent took 37.083 seconds to execute all the problems.

The algorithmic complexity of my AI agent is O (n^4) where n is the number of options.

Second Submission

The second submission was sent on 2019-10-27 19:40:42 UTC. I started from scratch for this submission as the first submission approach hasn't been very fruitful. In this submission, I implemented dark pixel density scorer (similar to white to dark pixel ratio scorer in 2x2 RPM solver). For figure comparison, I first convert all the pixels of each image into either white or black pixel. Using, dark pixel density percentage, I determine the percentage of black pixel to all the pixels in the same image. After calculating, the dark pixel density for each figure, I compare dark pixel density of a figure to other figures in the same row and in the same column. This helps us in establishing if the dark pixel is decreasing/increasing/constant across rows/columns. If the pixel density is found to be increasing across all the rows, we compare fig H with all the 8 options and give score value of 1 to all the options that have a higher dark pixel density than fig H. Similarly, if the pixel density is found to be constant across all the columns, we compare fig F with all the 8 options and give score value of 1 to all the options that have the same dark pixel density as fig F with a comparison error threshold of 0.1. This scoring mechanism is similarly performed for all columns and all rows each for decreasing/increasing/constant cases. If we find that there is an inconsistency, like 1 row (or column) follows constant/increasing but the other row (or column) is decreasing, we assign a score of 0 to all the options because we don't know what is the exact pattern. Also, it is perfectly fine to have a relationship where say dark image pixel density across rows is increasing but pixel density across column is decreasing (as rows and columns relationships are independent). The score for each option after establishing horizontal relation and vertical relationship are added with the same weightage.

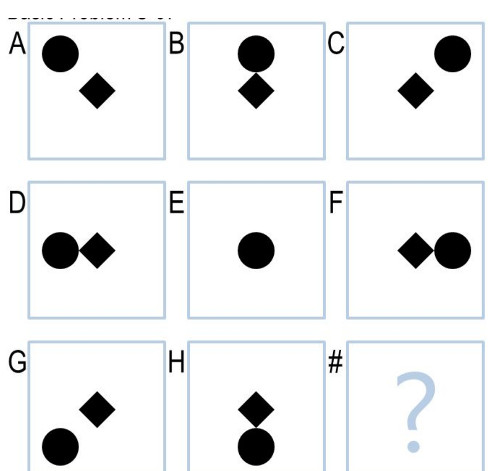
The approach of AI agent in this submission is not at all similar to a human's approach because, as humans, we don't look at dark pixel density percentage for each figure to find out possible relationships between figures. We instead look at the figures overall (and not in terms of pixels).

Performance:

Problem Set C: Basic problems: 5/12 ; Test problems: 7/12; Ravens problems: 5/12; Challenge problems: 2/12. My AI agent significantly improved in performance. It solved almost all the problems that had a unique option with increasing dark pixel density. To be exact, the AI agent was able to solve Basic Problem C-01, Basic Problem C-03, Basic Problem C-04, Basic Problem C-05, Basic Problem C-06. It failed on all the problems, that didn't have dark pixel density relationship. In some cases, I saw my agent giving identical score to multiple option because multiple options can satisfy relationships based on just dark pixel density. In terms of **efficiency**, my agent took 12.627 seconds to execute all the problems. The algorithmic complexity of my AI agent is O (n) where n is the number of options.

Third Submission

The third submission was sent on 2019-10-27 21:30:58 UTC. In my third submission, I implemented the reflection scorer as I found reflection property in the problem Basic Problem C-07. Basic Problem C-07 is not entirely a reflection problem but still can be solved based on the partial reflection property. To solve this problem, my agent checks if (fig A and fig G) and (fig B and fig H) have top down reflection relationship. Similarly, my agent also check if (fig A and fig C) and (fig D and fig F) have left right reflection relationship. We ignore the relationship of center with any other figure. If any vertical top down reflection relationship is found, my agent does a top down reflection transformation on figC and compares with it all the options. The options that match with the error threshold of 0.07 are given a score of 1, 0 otherwise. Similarly, if horizontal left right reflection relationship is found my, agent does, a left right reflection transformation on fig G and compares with all the options. The options that match with the error threshold of 0.07 are given a score of 1, 0 otherwise. I also made some threshold modification in dark pixel comparison to make it more strict which solved Basic Problem C-10 that is based on dark pixel density.



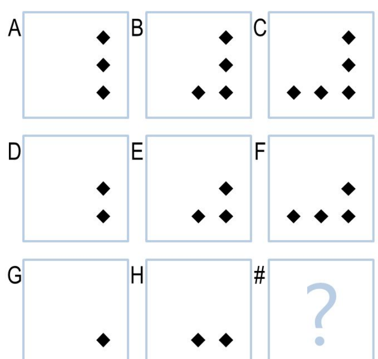
The approach of AI agent to determine reflection relationship in this submission is similar to a human's approach because, as humans, we can quickly see if one figure is top down/left right reflection of another and can answer the 3x3 RPM problem quickly.

Performance:

Problem Set C: Basic problems: 7/12 ; Test problems: 9/12; Ravens problems: 6/12; Challenge problems: 2/12. Problem Set B: Basic problems: 12/12 ; Test problems: 9/12; Ravens problems: 8/12; Challenge problems: 7/12. With this change, my agent's score improved significantly on Test and Ravens. The reflection logic solved Basic Problem C-07 and threshold modification solved Basic Problem C-10. In terms of **efficiency**, my agent took 18.397 seconds to execute all the problems. The increase in the time it took as compared to Second submission is expected because of the addition of the reflection scorer. The algorithmic complexity of my AI agent is O (n) where n is the number of options.

Fourth Submission

The fourth submission was sent on 2019-10-27 23:28:52 UTC. In my fourth submission, I solved Basic Problem C-11, by implementing diff image scorer. Diff image scorer, subtracts one image from another within the same row/column. The idea here is to compare the dark pixel pixel density of the subtracted image. If the dark pixel density of the subtracted image is same across the rows, we can make the conclusion that, there was an addition of the same shape across all rows. Similarly, we also compute dark pixel pixel density of the subtracted image in each column and then compare it with other columns. Eg: In Basic Problem C-11, we subtract fig A from fig B and compare the dark pixel density of the subtracted image (fig B - fig A) with the dark pixel density subtracted image (fig C - fig B). If we find the same relationship between subtracted image (fig E - fig D) and the subtracted image (fig F - fig E), this would mean that row is follow a relationship of addition of some figure. Therefore, we calculate dark pixel density of (option-figH) for each option and compare it with dark pixel density of (figH - figG), if they are found to be similar with error threshold of 0.1, we give that option, a score of 1, 0 otherwise. This is also followed in a similar fashion across columns. Note that, this Diff image



scorer is only called if the rows/columns have consistent dark pixel density increasing/decreasing order.

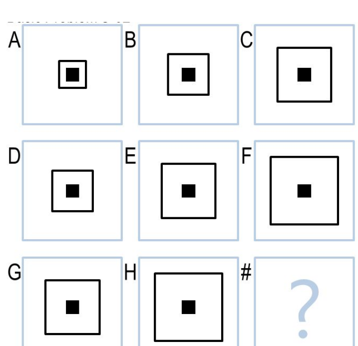
The approach of AI agent to determine subtraction image relationship in this submission is somewhat similar to a human's approach because, as humans, we instead do addition of one figure on another for this kind of problems. However, we can think of it as opposite sides of the same coin because for humans image addition is way simpler, but at the end though, both subtraction and addition is doing the exact same thing.

Performance:

Problem Set C: Basic problems: 8/12 ; Test problems: 10/12; Ravens problems: 6/12; Challenge problems: 3/12. Problem Set B: Basic problems: 12/12 ; Test problems: 9/12; Ravens problems: 8/12; Challenge problems: 7/12. With this change, my agent's score improved on all the problems. The diff image scorer logic solved Basic Problem C-11. In terms of **efficiency**, my agent took 15.085 seconds to execute all the problems. The decrease in the time it took as compared to Third submission is expected because of some efficiency optimization, I did to my Agent. The algorithmic complexity of my AI agent is O (n) where n is the number of options.

Fifth Submission

The fifth submission was sent on 2019-10-28 00:21:26 UTC. In my fifth submission, I was able to solve Basic Problem C-02. To solve this problem, I implemented dark pixel increment scorer. The idea here is to look at the dark pixel increment value from fig C to fig F and fig G to fig H. Here, I compare the dark pixel density of fig F, dark pixel density of fig C, dark pixel density of fig H, dark pixel density of fig G. For row comparison, I calculate dark pixel density of fig H - dark pixel density of fig G and compares it with each of the dark pixel density of options - dark pixel density of H. If they are found to be similar with error threshold of 0.2, my agent gives a score of 1, 0 otherwise. Similarly, for column comparison the agent calculates dark pixel density of fig F - dark pixel density of fig C and compares it with each of the dark pixel density of options - dark pixel density of F with an error threshold of 0.2. If there is a match, the agent gives



a score of 1, 0 otherwise. Note that this scorer is only called in case of strictly increasing/decreasing relationships across rows/columns.

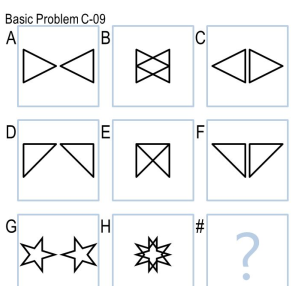
The approach of AI agent to determine dark pixel increment scorer in this submission is not at all similar to a human's approach because, as humans, we don't look at dark pixel density exact increment in terms of a metric to determine if the next sequence of the figure also has a similar increment in a row/column.

Performance:

Problem Set C: Basic problems: 9/12 ; Test problems: 9/12; Ravens problems: 9/12; Challenge problems: 4/12. Problem Set B: Basic problems: 12/12 ; Test problems: 9/12; Ravens problems: 8/12; Challenge problems: 7/12. With this change, my agent's score improved significantly on Ravens problems. The dark pixel increment scorer logic solved Basic Problem C-02. In terms of **efficiency**, my agent took 15.217 seconds to execute all the problems. The slight increase in the time as compared to Fourth submission is expected because of the addition of this new logic of dark pixel increment scorer. The algorithmic complexity of my AI agent is O (n) where n is the number of options.

Sixth Submission

The sixth submission was sent on 2019-10-28 03:01:14 UTC. In this submission, I



was able to solve Basic Problem C-09. To solve this problem, I added split/slice and compare scorer to my AI agent. My agent slices figures A, C, D, F and G vertically into two halves. The middle column is not taken into account at all. After slicing each image, the comparison is made across the rows. Left half of the figures in the first column is compared to the Right half of figure in the same row but in the third column and similarly right half of the figures in the first column is compared to the left half of the figure in the same row but in the third column. Eg left half of fig A is compared with right half of fig C and right half of fig A is compared with left half of fig C. This comparison is followed in both of the top two rows. If the relationship is followed in both the rows, fig G is sliced vertically into two halves and all the options are also sliced vertically into two halves. Left half of fig G is compared to right half of each of the options. and similarly, right half of fig G is compared to left half of each of the options. If an exact match is found for both halves, the option is given a score of 1, 0 otherwise.

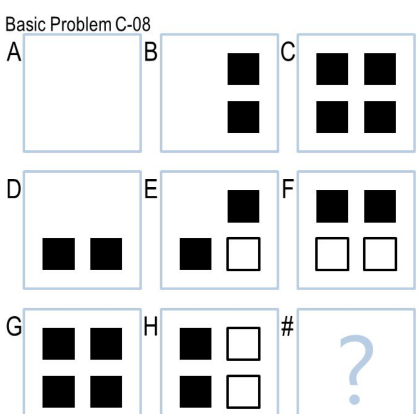
The approach of AI agent to determine similarity by splitting the image into two halves vertically is not at all similar to a human's approach because, as humans, we don't cut, the images and compare pixel by pixel. If a human solve this problem, he/she would see that the two shapes inside any of the figure in column 1 are moving towards each other. Column 2 shows them meeting. Column 3 shows the figures crossing each other and leaving in the opposite direction.

Performance:

Problem Set C: Basic problems: 10/12 ; Test problems: 10/12; Ravens problems: 9/12; Challenge problems: 4/12. Problem Set B: Basic problems: 12/12 ; Test problems: 9/12; Ravens problems: 8/12; Challenge problems: 7/12. With this change, my agent's score improved significantly on Set C: Basic problems as it is now able to solve Basic Problem C-09. Since, I don't have access to Test and Ravens problems, I am not sure where my agent is struggling at. In terms of **efficiency**, my agent took 16.182 seconds to execute all the problems. The slight increase in the time as compared to Fifth submission is expected because of the addition of this new logic of split/slice and compare scorer. The algorithmic complexity of my AI agent is O (n) where n is the number of options.

Seventh Submission

The seventh submission was sent on 2019-10-28 04:06:52 UTC. In this submission, I was able to solve Basic Problem C-08. I wasn't able to solve this problem with the exact correct approach, I solved it by a slightly different approach by implementing a dark pixel decrement scorer (similar to the one we used in submission 5). The idea here is to only look at the (fig G and fig H) and (fig C and fig F) and ignore other figures. Basic idea that I implemented is to look at the dark pixel decrement value from fig C to fig F and fig G to fig H. For row comparison, I calculate dark pixel density of fig H - dark pixel density of fig G and compares it with each of the dark pixel density of options - dark pixel density of H. If they are found to be similar with



error threshold of 0.2, my agent gives a score of 1, 0 otherwise. We do this similarly for column as well.

The approach of AI agent to determine dark pixel decrement scorer in this submission is not at all similar to a human's approach because, as humans, we don't look at dark pixel density exact decrement in terms of a metric to determine if the next sequence of the figure also has a similar decrement in a row/column.

Performance:

Problem Set C: Basic problems: 11/12 ; Test problems: 10/12; Ravens problems: 9/12; Challenge problems: 4/12. Problem Set B: Basic problems: 12/12 ; Test problems: 9/12; Ravens problems: 8/12; Challenge problems: 7/12. With this change, my agent is now able to solve Basic Problem C-08. Since, I don't have access to Test and Ravens problems, I am not sure where my agent is struggling at. In terms of **efficiency**, my agent took 16.182 seconds to execute all the problems. The slight increase in the time as compared to Fifth submission is expected because of the addition of this new logic of split/slice and compare scorer. The algorithmic complexity of my AI agent is O (n) where n is the number of options.

Eighth Submission

The eight submission was sent on 2019-10-28 04:28:49 UTC. In this submission, I added a guesser logic to answer questions on which my agent wasn't able to determine an answer.

This submission with a change on guessing answer is very similar to how humans approach, when they are unsure of what is the right answer. We guess.

Performance:

Exact same score as the seventh submission. My AI agent is still struggling with problem Basic Problem C-12 where we increase black square one at a time. In terms of **efficiency**, my agent took 16.098 seconds to execute on all the problems which is very similar to the seventh submission. The algorithmic complexity of my AI agent is O (n) where n is the number of options.

Conclusion: I started this project project with a trial and error approach of submission 1 to see if the 3x3 RPM problems if divided into multiple sub 2x2 problems can be solved by my AI agent or not. However, after the first submission, all the subsequent submission have been done with deliberate improvement. In all of my implementation and submissions, I have incrementally added independent logic to improve my agent by targeting one problem at a time.

My AI agent for is not similar to human at all in respect to solving 3x3 RPM problems. My AI agent uses multiple scoring mechanisms which are not very intuitive to human way of solving the 3x3 RPM problems. My AI agent solves every problem by doing some pixel counting, manipulation, subtraction, dark pixel increment, dark pixel decrement, slicing images into halves which is not human like at all. The two scorer mechanism used by AI agent which are somewhat human like are diff image scorer which calculated diff images and reflection scorer. For diff image scorer, my AI agent differs slightly from human as it's diff image scorer which computes subtraction image and compares it with other subtraction images. As a human, I instead do addition of one figure on another for this kind of problems. However, we can think of it as opposite sides of the same coin because for humans image addition is way simpler, but at the end though, both subtraction and addition is doing the exact same thing.

If I had more time, I would have tried to use edge detection techniques, and would have tweaked image equality threshold to see if those help with the Test & Ravens problems. I would have also tried to come up with logic to solve Challenging problems. I would have also explored verbal approach in which my agent would also take verbal input and would score options based on both visual and verbal data.