CS 7637 Project 1 Reflection:

Abhijeet Gaurav
agaurav@gatech.edu

Introduction

This journal covers the design and implementation of my AI agent that solves the 2x2 Raven's Progressive Matrices (RPM) problems. My agent will only use visual representation as an input to my AI Agent. Horizontal/Vertical transformation: My agent will do a transformation on A, compare it with B to see if there is a horizontal relationship; Similarly, my agent will do a transformation on A & will compare it with C to see if there is a vertical relationship. If horizontal relationship is found, same transformation on C will be applied and then compared to 6 options. If vertical relationship is found, transformed B will be compared with all the options. Each option will be assigned a score. Option with the highest score will be agent's answer. Applied transformations can either be trivial (rotation/identical/reflection) or non-trivial (difference of difference image/fill image with color transformation). Trivial relation will mean that two images are in fact the same but only differ in terms of rotation and reflection. If no trivial relationship exists, agent will then apply non-trivial transformation, such as scoring and comparing after filling image with color and scoring by comparing difference image that I am going to talk more in detail below.

Comparison	Description
Identical +	Check if A&B, A&C, B&C are identical. (special case of 0° rotiation)
Rotation +	Rotate one fig by 90°, 180°, 270° and check if it becomes identical to other 2
Reflection +	Take top-down/left-right reflection and check if this image is identical to other 2
Pixel ratio of white/black *	Compare figures based on ratio of white/black pixel
Score diff Image φ	Check if diff images are identical (eg: A-B & C-options is identical)
Fill figure scorer φ	Fill figures with color and score

✱ Decides if the image falls into trivial or non-trivial transformation case. (Blue)

φ Non-trivial transformation case (Red)

+ Trivial transformation case (Green)

First Submission

The first submission was sent on 2019-09-15 06:53:23 UTC. The first task of my AI agent is to build possible relationships between figures. For figure comparison, I first convert all the pixels into either white or black pixel. This pixel conversion into one of the two colors allows my agent to measure if the white to black pixel ratio between two figures is identical. The idea here is to make a smart and robust tester to reason on figure similarity. The second task of my AI agent is to analyze what transformations to consider. Since, in 2x2 problem, there can be a transformation from A -> B and/or A -> C, my agent compares A with both B and C in terms of white to black pixel ratio. If the pixel ratio between two figures are found to be at least 99% similar, my agent makes the assumption that these figures are the same and only differ in terms of orientation (rotation/reflection) and decides to proceed with the trivial transformations (rotation and reflection). Otherwise, it considers the transformations to be non-trivial (and applies diff image transformation). My agent has different modes on what transformations to generate and in which direction (horizontal and/or vertical). The idea here is to make a smart Generator that only generates optimal transformations based on how likely is that the generated transformation will ultimately be useful (exact match with other figures). My agent's generator intelligence is decided based on mode number which comes from white to black pixel ratio. The modes that my agent uses are defined in the table below (Green ones are trivial; Red one is non-trivial):

Modes	Description
Mode 0	Establish trivial transformation in both horizontal and vertical direction
Mode 1	Establish trivial transformation in horizontal direction
Mode 2	Establish trivial transformation in vertical direction
Mode -1	Establish non-trivial transformation relation as A,B and C are not similar

Rotation: A is rotated by 0º (identical case), 90º, 180º and 270º and then compared it with B and/or C based on the mode value. If B/C is found to be the exact match with the rotated A with say rotation value x, my agent's tester records first rotation value for which the images were found to be similar. It then rotates C/B with the same rotation x and then compares it with all the possible options [1, 2, 3, 4, 5, 6]. All the options which were exact match (atleaset 96% equal) as rotated C/B are assigned a score of 1, otherwise 0. **Reflection**: Similar to rotation, AI agent follows the same behavior of forming reflection relationship between A, B

and C based on the mode value. The reflection transformation does both the top down reflection and left right transformation on A and compares it with B/C based on the mode value. Each option is scored based on whether it is exactly matched (at least 96% equal). If there is an exact match, the option is given a score of 1, 0 otherwise. **Pixel scoring:** My agent also has a pixel score, which has info on the similarity based on white to black pixel ratio. This score makes sure that the options that are similar in terms of white to black pixel ratio to the transformed figures get a higher score. This score eliminates all those options which have all the black colored pixels that might have been incorrectly selected by Reflection scorer or Rotation scorer. **Final scoring for trivial transformations:** In the final scoring, I scale pixel score by 5, reflection score by 4 and add them with the rotation score. This scoring solves all the Basic 2x2 problems from problem 1 to problem 8. ***Non-trivial transformation:*** The non-trivial transformation (i.e. diff Image scorer) is only triggered if the AI agent is not able to find any answer from trivial transformation or if the trivial transformation gives the same score for all the options. The goal of diff image scorer is to compare subtracted image. 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. If the subtract image (A-B)/(A-C) is found to be exact match with any of the (C-option)/(B-option), this particular option gets a score of 1; 0 otherwise. Through this approach, my diff image scorer, was able to solve Basic 2x2 problems from problem 10 to problem 12.

Agent comparison to humans approaching the problem: I think my agent approaches exactly as I (or any other human) approaches on the set of problems it is successfully working on. My AI agent follows the exact same steps that I follow to solve an 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. My AI agent's pixel scorer, which makes sure that in a trivial transformation, the agent eliminates all the options that have different white to black pixel ratio as compared to the A/B/C. I do the same thing when I am looking at an RPM problem, I quickly rule out those options that I know are not going to be the solution because they have different figures that are not related to the problem (The basic building block of different images are distribution of black and white pixels which my agent sees). However, something different about my AI agent from me is that, it is very mathematical in

nature. I have to tell my AI agent explicitly to ignore 4% transformation error or 1% similarity error. To me those errors are not even there because the human eye is not that precise in recognizing these small errors.

Performance: Basic problems: 11/12 ; Test problems: 9/12; Ravens problems: 8/12; Challenge problems: 5/12. My agent performed very well on almost all the Basic problems but 1 in its first iteration. Since, Test problems and Ravens problems are kind of black box, I am not entirely sure on what 3 types of test problems, my agent is currently failing on. The only basic problem that it could not solve is problem 9, because this problem requires color filling in shapes and there is no logic in my agent to solve this kind of problem. I would try to solve this problem in my future iteration. In terms of **efficiency**, my agent took 9.52826 seconds to execute all the problems. The algorithmic complexity of my AI agent is O (n) where n is the number of options.

Second Submission

The second submission was sent on 2019-09-17 07:50:25 UTC. My AI agent only scored 9/12 on test problems. Since, these are black box test cases, I don't know what exactly is going wrong. In this version, I made my rotation code more robust and refactored some of the methods to consistently use image equality error threshold of 3.7% for comparing if the two images are exact match and made some changes on the pixel density scorer to make it more strict. In my rotation code from first submission, while doing rotation transformation on A step by step and comparing it with say C, if the lowest rotation transformation was selected, the other possible rotation transformation used to get ignored because they were not even considered as I used to break out of the 'for' loop. Now, in the current new submission, my agent builds up a list of probable rotation transformations based on the comparison between rotated A and B/C and considers each rotation individually; and applies the same rotation one at a time to C/B and then compares each rotated C/B with the options. I made this change to improve the correctness of the rotation scores that might be impacting test problem scores.

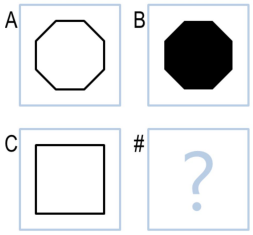
The changed AI agent makes it even more human like in approaching RPM problems and is similar to how humans would approach the rotation problem because, if a 90^{o} rotation doesn't bring any fruitful result, we'll try to do 180^{o} and 270^{o} rotations too, to make sure we are not neglecting any possible solution.

Performance

Basic problems: 11/12 ; Test problems: 9/12; Ravens problems: 8/12; Challenge problems: 6/12. Contrary to what I thought, this rotation change didn't make any progress on test problems but overall change which encompasses change in image equality error threshold did solve Challenge Problem B-05. In terms of **efficiency**, the AI agent took 9.53098 seconds to execute on all the problems. The increase of 2.72 ms from submission 1 is expected because of added extra rotations that the AI agent has to consider. The algorithmic complexity of my AI agent is still O (n) where n is the number of options.

Third Submission

The third submission was sent on 2019-09-22 02:42:09 UTC. In this submission, I added logic to my AI agent so that it can address problems based on coloring such as basic problem (Basic Problem B-09). The exact problem is below:



For this kind of problem, my AI agent detects the shape in each of the figures to determine if the figures are enclosed. Note that this is a non trivial transformation and it is only executed if the figures don't show any trivial relationship. Based on the white to dark pixel ratio, my agent decides which figure it should fill up with black color. AI agent picks the figure with higher white to black pixel ratio, does a black color filling transformation by filling the shape inside the figure with black color. It then compares this transformed black color filled image with the rest of the lower white to black pixel ratio figures to determine if there is an exact match. If an exact match is found, the agent comes to know that the transformation is reasonable. It then does a similar transformation to either all the options or the third figure depending on their white to black pixel ratio and then compares them as well to find out if they are of the exact match. If there is an exact match, AI agent's fill and score method assigns 1 as the score, 0 otherwise.

The approach of AI agent is very similar to how other humans/I would approach the color filling problem. I would see at all the options and will quickly know which one of them is filled with black color. My agent does the same thing, it fills the shapes in figures with black color and then compares the option one by one.

Performance

Basic problems: 12/12 ; Test problems: 9/12; Ravens problems: 8/12; Challenge problems: 6/12. My AI agent is now able to solve all the basic problems achieving a perfect score. However, I am still not sure why scores on test problems and Ravens problems didn't increase. Because test and raven's problems are invisible, I am not sure what kind of problems, my agent is struggling with. In terms of **efficiency**, the AI agent took 12.5754 seconds to execute on all the problems. The increase of 3 seconds from submission 2 is expected because of the addition of black color filling logic and related transformations that the AI agent has to consider. The algorithmic complexity of my AI agent is still O (n) where n is the number of options.

Fourth Submission

The fourth submission was sent on 2019-09-22 03:24:58 UTC. In this submission, I added more modes. Previously, my AI agent only considered horizontal or vertical relationship or a combination of the two. In this submission, I added more modes, so that my agent can also consider diagonal relationships. My agent now does all the trivial and non-trivial transformation on B and then compares it with C. If the resulting image of the transformed B matches C, my agent does similar transformation to A and then compares transformed A to all the options. The options that exactly match transformed fig A are given the same score as in case when we have horizontal and vertical relations. In this submission, I also tweaked thresholds to make them more lenient for image comparison and image similarity in terms of white to black pixel ratio. I decided to consider diagonal relationship to see if the Test problems have such kind of relationship on which my agent might be failing. I tweaked the thresholds for image comparison and image similarity because some of the challenging problems do not have the same pixel alignment.

My AI agent's making diagonal relationship is similar to how I/other humans would approach RPM problem. If in an RPM test, I see a diagonal relationship which is way simpler, I take that approach. My agent does the exact same thing.

Performance

Basic problems: 12/12 ; Test problems: 9/12; Ravens problems: 8/12; Challenge problems: 7/12. Forming diagonal relationship did not help at all. My agent gave the exact same score for the test problems. However, tweaking some of the image comparison and similarity threshold did lead to solving 'Challenge Problem B-10' which deals with rotation. Because test and raven's problems are invisible, I am not sure what kind of problems, my agent is struggling with. In terms of **efficiency**, the AI agent took 12.067 seconds to execute on all the problems. The decrease of 0.5 seconds from submission 3 is expected because of some comparison improvements, I did to my AI agent. The algorithmic complexity of my AI agent is still O (n) where n is the number of options.

Fifth Submission

The fifth submission was sent on 2019-09-23 11:10:52 UTC. In this submission, I mainly tried to approach the problem via trial and error approach and made image equality thresholds for reflection scorer, rotation scorer, diff image scorer, fill image scorer a lot liberal. I decided to make these changes, so that can see if some of the test problems which are failing are like challenging problems which are very much dependent on these thresholds. This is because the images in options do not have same pixel distribution as A/B/C even if they look the same.

This version of AI agent is even more like humans because it marks two figures to be equal even though they have a difference of 8%. Humans can't differentiate between two figures even if their pixel distribution is slightly different. This is because the human eye is not that precise in recognizing minute pixel detailed difference between figures. We look at the whole picture and and don't think of image in terms of pixels.

Performance

Basic problems: 12/12 ; Test problems: 8/12; Ravens problems: 8/12; Challenge problems: 7/12. Changing thresholds didn't help at all as I got almost the same score as before. Since, I can't see Test and Ravens problems, I am not sure on the exact problems on which my agent struggled. My agent still gave a perfect score on Basic problems. In terms of **efficiency**, the AI agent took 12.0599 seconds to execute on all the problems which is almost similar to fourth submission.The algorithmic complexity of my AI agent is still O (n) where n is the number of options.

Conclusion

Before the first submission, locally I designed my AI agent through deliberate improvement. For each kind of problem, I spent a lot of time thinking about how as a human I approach a particular RPM problem and then came up with the idea/algorithm to solve each kind of problem. In that respect, I agree that I targeted one type of problem at a time. Also, the AI components such as rotation scorer, reflection scorer, pixel scorer, image diff scorer, image fill scorer are made in phases to target different type of problem shows the deliberate improvement that I did locally. However, after my first submission, all the subsequent submissions except the third submission, has been done with trial and error approach as I wasn't sure on what test problems, my agent was struggling with. Although, I haven't been able to increase score beyond 9/12 in test problems, each of the submissions was better than the previous submission, as they improved challenging problem scores and AI agent in general in the areas which do not have test cases.

My agent is very similar to how a human would approach the test. The agent is designed by me (a human) and approaches the problems in the same way as I do. Before starting to code the logic, I spent a lot of time thinking myself by slowing down and thinking deeply, what steps I am taking when I approach a particular problem, and I made sure my agent approaches the problem in the same way. Example, in the problems related to reflection and rotation, my agent approaches the problem in the same way as I do. It tries to rotate/reflect and then compare. It gives more weightage to the reflection than rotation which I also do. Another example where my AI agent differ slightly from human is 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, corners counting and would have tweaked image equality threshold to see if those help with the Test 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.