Report of Project 3

ZHENG FU

zfu66@gatech.edu

Abstract. In this report, I describe the algorithms underlying a knowledge-based AI agent for solving 3 x 3 puzzles in Raven's Progressive Matrices test sets D and E using visual representations. A further discussion was made to elucidate the difference between the AI agent behaviors and the ways a human using to solve these problems.

Section I

This project is a succeeding work of Project II in which I used visual representations to solve the 3 x 3 Raven's Progressive Matrices (RPM) test set C. Recall that in Daniel R. Little et al. 2012 paper there are eight basic rules to solve the RPM problem visually:

> "1) constant, 2) increment or 3) decrement, 4) permutation, 5) logical AND, 6) logical OR, 7) logical XOR and 8) a Distribution of 2 rule." (Little 2012)

In Project II, my attentions were on constant rule, increment rule, decrement rule, logical AND rule and logical OR rule. And it is obvious that they are not enough to solve the problems in RPM test sets D and E. Therefore, I intend to implement permutation rule, logical XOR rule and a Distribution of 2 rule in this project, here are the steps I improved my AI agent over time:

1. On the first round I designed the AI agent to check the simplest rules: "Identical" rule, "Add" rule, "Increment/Decrement" rule, "2x" rule and "Permutation" rule.
2. On the second round the AI agent was mainly trained by the problems in basic RPM test set D, and was able to check the following rules: 1) "Horizontal Add" rule, 2) "Vertical Add" rule, 3) "Permutation Add" rule and 4) "Permutation 2x" rule.

3. On the third round the AI agent was mainly trained by problems in basic RPM test set E and was able to check the following rules: 1) “Shift Blend” rule, 2) “Flip Shift Blend” rule, 3) “Double Blend” rule and 4) “XOR” rule.

All these rules will be explained in detail on Section II.

Section II

As far as my final agent was concerned, its process of solving the problems contains three major parts: 1) Pre-processing the images, 2) Taking actions on images and 3) Determining the correct answer through a series of rules.

1. Pre-processing the images.

All images were first resized to 184 pixels times 184 pixels and converted into greyscale (“L”) as Numpy arrays. Then if a pixel grey scale value is no more than 65, it will be reset to 0 (black), otherwise it will be reset to 255 (white). This step will make the image sharper and reduce the errors while computing the pixel differences. Finally the greyscale Numpy arrays were transformed to python Pillow image objects.

2. Taking actions on images.

There are seven major actions that my final AI agent often takes on the images:

1). Blend.

The AI agent will overlay two images and generate a blended image. Note that the blended figure should be normalized, since the un-overlapped regions are much lighter than the overlapped regions. Unlike the normalization in pre-processing step, if a pixel grey scale value is no more than 200, it will be reset to 0 (black), otherwise it will be reset to 255 (white). *PIL.ImageChops.blend(image1, image2, alpha)* function was used to perform this operation and the alpha value was set to 0.5.

2). Invert.

The AI agent will swap the white pixels and black pixels in an image. *PIL.ImageChops.invert(image)* function was used to perform this operation.

3). Flip.

The AI agent will flip the image vertically. *PIL.ImageOps.flip(image)* function was used to perform this operation.

4). Shift-Blend.

The AI agent shifts image1 -45 pixels on x-axis and shifts image2 35 pixels on x-axis. Then the agent blends shifted image1 and image2. The "Shift" operation was performed by *PIL.ImageChops.offset(image, xoffset, yoffset=None).*

5). Flip-Shift-Blend.

The AI agent shifts image1 -26 pixels on x-axis and flips image2 followed by shifting it 26 pixels on x-axis. Then the agent blends shifted image1 and image2.

6). XOR.

Both image1 and image2 were inverted and then blended. The blended image is inverted to generate the final XOR image.

7). Compute the pixel difference.

The AI agent will compute the absolute value of the pixel-by-pixel difference between the two images. *PIL.ImageChops.difference(image1, image2)* function was used to perform this operation. And pixel difference percentage was defined as the pixel-by-pixel difference between the two images over the total pixels (184 x 184 = 33856).

2. Determining the correct answer by rules.

The AI agent employs a series of rules one-by-one until it finds the correct answer. If all rules were applied and no answer was selected, the AI agent will skip such questions.

1). "Identical" rule.

If the pixel difference percentage between two images is no more than 1%, the AI agent will believe these two images obey "Identical" rule and they are "Identical".

2). "Add" rule.

Image 1 and 2 are blended and if the pixel difference percentage between blended image and image 3 are no more than 1%, the AI agent will believe these three images obey "Add" rule.

3). "Increment/Decrement" rule.

Image 1 and 2 are blended and the pixel-by-pixel difference is computed between blended image 1&2 and image 3 (pixel-difference-123). Image 4 and 5 are blended and the pixel-by-pixel difference is computed between blended image 4&5 and image 6 (pixel-difference-456). If the ratio between pixel-difference-123 and pixel-difference-456 is no more than 1.05 and no less than 0.95, the AI agent will believe these six images obey "Increment/Decrement" rule.

4). "2x" rule.

If the number of black pixels of image 1 plus that of image 2 is equal to the number of black pixels of image 3 times two, the AI agent will believe these three images obey "2x" rule.

5). "Permutation" rule.

If image A and image E, image B and image F, image C and image D, image F and image G, image D and image H are all following "Identical" rule, then the AI agent will believe it could use "Permutation" rule to figure out the problem. The correct answer and figure E should obey the "Identical" rule.

6). "Horizontal Add" rule.

In the following four cases, the AI agent will believe the problem could be solved by "Horizontal Add" rule.

1. Image A, B and C obey "Add" rule. Image D, E and F obey "Add" rule. Thus Image G, H and answer should obey "Add" rule as well.

2. Image B, C and A obey "Add" rule. Image E, F and D obey "Add" rule. Thus Image H, answer and G should obey "Add" rule as well.
3. (Image A, B, C) and (Image D, E, F) obey "Increment/Decrement" rule. Thus (Image D, E, F) and (Image G, H, answer) should obey "Increment /Decrement" rule as well.
4. (Image B, C, A) and (Image E, F, D) obey "Increment/Decrement" rule. Thus (Image E, F, D) and (Image H, answer, G) should obey "Increment /Decrement" rule as well.

7). "Vertical Add" rule.

In the following four cases, the AI agent will believe the problem could be solved by "Vertical Add" rule.

1. Image A, D and G obey "Add" rule. Image B, E and H obey "Add" rule. Thus Image C, F and answer should obey "Add" rule as well.
2. Image D, G and A obey "Add" rule. Image E, H and B obey "Add" rule. Thus Image F, answer and C should obey "Add" rule as well.
3. (Image A, D, G) and (Image B, E, H) obey "Increment/Decrement" rule. Thus (Image B, E, H) and (Image C, F, answer) should obey "Increment /Decrement" rule as well.
4. (Image D, G, A) and (Image E, H, B) obey "Increment/Decrement" rule. Thus (Image E, H, B) and (Image F, answer, C) should obey "Increment /Decrement" rule as well.

8). "Permutation Add" rule.

In the following two cases, the AI agent will believe the problem could be solved by "Permutation Add" rule.

1. Image A, F and H obey "Add" rule. Image E, G and C obey "Add" rule. Thus Image B, answer and D should obey "Add" rule as well.

2. (Image A, F, H) and (Image E, G, C) obey “Increment/Decrement” rule. Thus (Image E, G, C) and (Image B, answer, D) should obey “Increment /Decrement” rule as well.

9). “Permutation 2x” rule.

If image A, H, F obey “2x” rule and image E, C, G obey “2x” rule, the AI agent will believe the problem could be solved by “Permutation 2x” rule, and image D, answer, B should obey “2x” rule as well.

10). “Shift Blend” rule.

The AI agent will shift-blend image B and C and compare the result with image A, then shift-blend image E and F and compare the result with image D. If both pixel difference percentages are no more than 1%, the AI agent will believe the problem could be solved by “Shift Blend” rule. It means the pixel difference percentage between shift-blended (image H, answer) and image G is no more than 1%.

11). “Double Blend” rule.

If blended(image B and C) and blended(image A and C) obey “Identical” rule, and blended(image E and F) and blended(image D and F) obey “Identical” rule , the AI agent will believe the problem could be solved by “Double Blend” rule. It means the blended(image H and answer) is as same as blended(image G and answer) (pixel difference percentage <= 1%).

12). “Flip Shift Blend” rule.

If flip-shift-blended(image B and C) and image A obey “Identical” rule, and flip-shift-blended(image E and F) and image D obey “Identical” rule, the AI agent will believe the problem could be solved by “Flip Shift Blend” rule. It means the flip-shift-blended (image H and answer) is as same as image G (pixel difference percentage <= 1%).

13). "XOR" rule.

If XOR(image A and B) and image C obey "Identical" rule, and XOR(image D and E) and image F obey "Identical" rule, the AI agent will believe the problem could be solved by "XOR" rule. It means the XOR(image G and H) is as same as answer (pixel difference percentage <= 1%).

Section III

How many problems does it answer correctly?

The following tables summarize the accuracy of the final AI agent.

Table 1. The accuracy of the final AI agent for Problem Set D.

Problem Type	Correct	Skipped	Incorrect
Basic	12	0	0
Test	8	0	4
Challenge	2	0	10
Ravens	4	0	8

Table 2. The accuracy of the final AI agent for Problem Set E.

Problem Type	Correct	Skipped	Incorrect
Basic	12	0	0
Test	8	2	2
Challenge	2	8	2
Ravens	5	4	3

How efficient is it?

The AI agent uses 8.98002 seconds to solve the problems, so I believe it is quite efficient.

How general is it?

The AI agent did well on Basic problems yet for few Test problems and all Challenge and Ravens questions, its performance was not very good.

Different performance on the Basic and Test sets?

The AI agent has moderately different performance on Test set as Basic set. On Basic Problems D set, it has 100% accuracy and on the Test Problems D set it has 66.7% accuracy. On Basic Problems E set, it has 100% accuracy and on the Test Problems E set it has 66.7% accuracy.

Section IV

What types of problems does it currently answer incorrectly?

The final AI agent has 100% accuracy in both Basic Problems D set and E set. And it gives 4 wrong answers in Test Problems D set, skips 2 questions and gives 2 wrong answers in Test Problem E set, respectively. Due to the lack of information about Test sets, I do not know what types of problems in Test sets it skips and gives wrong answers.

What kinds of problems would it struggle with?

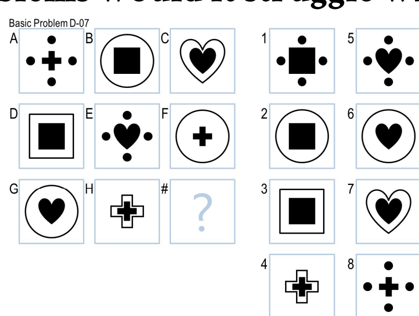

Figure 1. Basic Problem D-07.

According to the design of my final AI agent, there are two kinds of problems it will struggle with:

1. Problems like Basic Problem D-07. From Figure 1 we could see image C, D and H have the same pattern: one object is embedded into another object, and

these two objects have the similar shape. Thus it is difficult for the AI agent to separate these two objects and generate rules for making decisions.

2. Problems like Basic Problem E-09. From Figure 2 we could see each image has an upper object and a lower object. The AI agent should separate these two objects in each image and generate rules for the upper objects and lower objects, respectively. Yet it is hard for me to tell AI agent how to separate upper objects and lower objects accurately by rules.

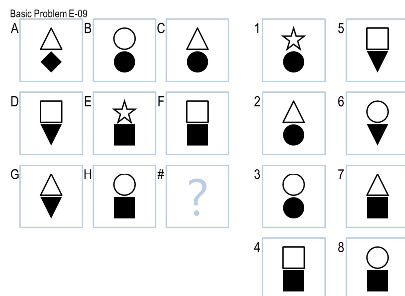

Figure 2. Basic Problem E-09.

Section V

For simple questions like Basic Problem E-01, the AI agent could "perceive" that two images are overplayed to generate the third image by computing pixel-by-pixel differences, as we human-beings directly see that by eyes. However for the following problems, I neither feel that the nature of my revisions reflecting the way a human learns from experiences, nor feel my final agent solves the problems similar to how a human would do so.

1. Basic Problem E-09: As discussed in Section IV, we human-beings could easily recognize the upper part and lower part in each image, and identify the pattern of upper parts on each row is "A-B-A", and that of lower parts is "A-B-A" hereby image 7 is the correct answer. The reason we are able to choose the correct answer is that we have prior knowledge of "Circle", "Square" and "Triangle" etc. In other word we make the decisions at the "object" level. Unlike human-beings, the AI agent employs a "divide-and-conquer" strategy

and makes decision at "pixel" level. For this kind of questions, it is quite hard for AI agent to mimic the way human perceiving and reasoning by using its "divide-and-conquer" strategy at "pixel" level.

2. Basic Problem E-12: For images on each row, we will intuitively flip the middle one first, then shift it and the right image to "proper" positions and compare the overplayed image with the left image. So how does the AI know where the "proper" positions are? Often we will setup a parameter for AI agent to control the shifting, and a slightly changing of such parameter may cause severe overfitting or underfitting. In other word, sometimes we have a fuzzy thinking and could get a mental picture, whereas AI agent always needs accurate thinking and is restricted by rules.

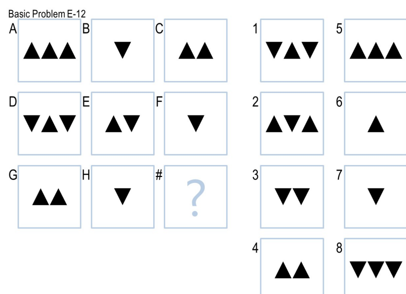

Figure 3. Basic Problem E-12.

References

1. Little, D., Lewandowsky, S., & Griffiths, T. (2012). A Bayesian model of rule induction in Raven's progressive matrices. *In Proceedings of the 34th Annual Conference of the Cognitive Science Society.* Sapporo, Japan.
2. Evans, T. G. (1964). A heuristic program to solve geometric-analogy problems. In *AFIPS '64 (Spring) Proceedings of the April 21-23, 1964, spring joint computer conference.* Washington, D.C., USA.
3. Joyner, D., Bedwell, D., Graham, C., Lemmon, W., Martinez, O., & Goel, A. (2015). Using Human Computation to Acquire Novel Methods for Addressing Visual Analogy Problems on Intelligence Tests. In *Proceedings of the Sixth International Conference on Computational Creativity*. Provo, Utah.