Jacob Kilver

Knowledge Based AI

Project 2 Reflection

3 July 2016

# Introduction

The purpose of this project was to build upon the previous project of solving 2x2 Raven’s Progressive Matrix (RPM) problems. In this project 3x3 RPMs were solved in addition to the 2x2 matrices from the last project. Generate and test was used to solve the 2x2 matrices, and a similar approach was applied here. However, since the patterns for 3x3 matrices were more varied, a more tailored approach to generate and test was used.

# Theory of Operation

In the previous project the “generate and test” method proved to be quite powerful but at times too general. As such, this same technique was applied here but with some modifications. The relationships between images and the patterns for the 3x3 RPMs were much more complex than in 2x2 matrices. (The 3x3 matrices were more than twice as large!) As such, generalizing to just a small set of relationships proved too complex to handle. The patterns were too varied to effectively abstract them, so the problems were handled on a case by case basis. With this structure the agent was able to answer problems it had encountered before, but it was not equipped solve new problems well unless these problems fit the same relationships it was trained to know.

Another difference between the 2x2 and 3x3 designs was how answers were selected. For the 2x2 matrices there were usually multiple ways to generate an answer that fit the pattern but would result in an incorrect answer. Some code was necessary therefore to handle cases where multiple generated guesses appeared in provided answers. For the 3x3 matrices with their complicated patterns meant that having multiple guesses that matched an answer was very unlikely. Therefore the first guess that appeared in the provided answers (within a certain confidence range) was returned as the answer to the RPM.

As with the 2x2 matrix problems, a purely visual approach for deriving solutions was taken here. While there were some problems that would have benefitted from a verbal approach, it was decided to remain with an entirely visual approach (for the sake of simplicity and due to time constraints). Note that this marks a deliberate difference from how the how the author reasoned while solving the RPMs. When initially solving the RPMs, the author found that he would derive a pattern and store this in some verbal form (i.e., shapes plus relationships between them). Then, when looking for an answer, a match to the verbal description would be sought ought, when entailed translating the visual representations of the answers into a verbal form. While all this happened rather quickly for the author, converting a visual scene to a verbal one was simply not within the scope of this project. While verbal representations were available, since half the problems will be without them in the next project, a strictly visual approach was taken here.

# Implementation

Python 2.7 was used to create the agent for this project. Since a visual approach was used, the Pillow library was used to manipulate the images.

As described above, the approach for this project consisted of a series of cases for generating an answer, usually specific to only one or two problems from the in-sample problem set. If the “guess” image that was generated matched reasonably well to one of the provided answers, that answer was assumed to be correct and the answer number was returned along with a confidence rating. As a result, if a match was made none of the other tests were performed.

Table 1 lists all the tests in order that the agent used to solve the RPMs for this project. It should be noted that some problems were deliberately not answered (though the agent might have given an incorrect answer). Also, sometimes the agent had some luck and correctly answered a problem that it was not intended to solve.

Table 1: Tests used by agent to solve RPMs

|  |  |  |
| --- | --- | --- |
| Test Description | Solution | RPMs it was intended to solve |
| Rows are all equal | Copy any image from last row | C-01, D-01 |
| Difference between columns is the same along a row | Apply difference between E and F to image H | C-11, C-12 |
| D and F are vertical reflections, B and H are horizontal reflections, C and G are 180° rotations | Rotate A 180° | C-07 |
| B and D are 90° CW rotations, C and G are equal, F and H are 90° CCW rotations | Rotate F, logical OR with H | C-08 |
| Pattern shifts right | Copy image A | D-02, D-03, D-11 |
| Pattern along rows, separate pattern that shifts right | Combine patterns | D-06 |
| Logical AND of all RPM question images | Take logical AND of all RPM questions (A-H) | C-05 |
| All the questions appear in the provided answers but one | Select the one answer that does not appear in the RPM questions | C-06 |
| Bisect image along vertical axis. Swap halves | Same as test | C-09 |
| Pattern along row, pattern along columns | Combine row and column pattern | D-04, D-05 |
| One pattern shifts left, another pattern shifts right | Combine right and left shift patterns | D-10 |

# Results and Discussion

The results of the in-sample testing can be seen in Figure 1 and Figure 2. When taken across all the in-sample problems (Basic and Challenge Sets C and D), the performance was as expected. Since the agent was only given knowledge about the Basic problems, performance was rather low on the Challenge problems (75% of these problems were answered incorrectly). However, since the agent was given detailed instructions on solving the Basic problems, performance on those questions was quite high; 75% of the Basic problems were answered correctly.

Figure : Accuracy of agent across Basic Problems Sets C and D and Challenge Problem Sets C and D

Figure : Accuracy of agent across Basic Problem Sets C and D

Regarding execution speed, no specific tests were performed. The author did note that the agent seemed slightly slower solving the 3x3 problems than when solving 2x2 problems. However this was expected since the 3x3 problems were larger and would require more time to solve. None of the tests in Table 1 were overly complicated and did not take that long to run. While it could take quite a while to run through all 11 tests, in practice all 11 tests were rarely performed.

The assumption that false positives would be rare did not hold for this implementation. Due to either small features or some of the imperfections between images, incorrect answers were quite common during the early stages of development. Fortunately an ordering of the tests was found so that the tests answered the specified problems correctly.

Part of the reason the false positives were so common was because the agent did not operate at the properly level of abstraction. It reasoned over pixels while humans reason over shapes. This was how the author reasoned as he solved the RPM. The placement of individual pixels did not matter nearly as much as the relative location of pixels (as they form shapes). The agent was not able to capture the relative location of the pixels and relied on a “dumb” testing mechanism to determine pixel-by-pixel how many pixels were the same between two images. As such this testing mechanism was highly susceptible to minor perturbations between images. Take for example Figure 3 and Figure 4 below. To the average human these two figures are identical, and for the purposes of answering an RPM they are. However, the circles are slightly offset, and when they are compared, as in Figure 5, there are some minor differences. (Note: Black denotes commonality, white denotes difference.) These minor differences were at times enough to confuse the agent as to what the correct answer was.

|  |  |
| --- | --- |
| C:\Users\Jacob\Documents\KBAI\Problems\Basic Problems D\Basic Problem D-03\A.png  Figure : Basic Problem D-03 Image A | C:\Users\Jacob\Documents\KBAI\Problems\Basic Problems D\Basic Problem D-03\E.png  Figure : Basic Problem D-03 Image E |

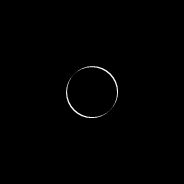


Figure : Difference between Basic Problem D-03 images A and E

# Conclusions and Future Work

In this project an agent to solve 3x3 RPMs was planned, developed, and tested. The performance of the agent depended mostly on the knowledge it was imparted with during development. For problems it was designed to answer the performance was acceptable, but for problems outside of that domain, the accuracy dropped quickly. This was primarily due to the difficulty in generalizing the 3x3 RPMs into a set of relationships that the agent could then use to solve new problems. Almost every RPM had a different relationship amongst the images, and creating a case for every possible combination simply was not feasible.

The largest shortcoming in the agent presently is the poor technique for comparing images. Currently this is a simple pixel-by-pixel comparison that is susceptible to noise. A new method needs developed that can determine the relationship between pixels in order to develop the concept of a shape, and then reasoning can be done over shapes rather than over pixels.