RPM Milestone Three

Allen Worthley

mworthley3@gatech.edu

Abstract—This document discusses the design and performance of an AI agent in RPM Milestone Three.

1 Design

This section discusses the inspirations of design, key calculations, and design of testing for the AI agent.

1.1 Standing on the Shoulders of Giants - Joyner's Agent 3

The AI agent described in this paper follows a similar methodology to Agent 3 described in Joyner's paper. The agent uses the key calculations Dark Pixel Ratio (DPR) and Intersection Pixel Ratio (IPR) to infer the selected image. Joyner et al (2015) defines DPR as:

> "the difference in percentage of the number of dark-colored pixels with respect to the number of total number of pixels in the continuous pixel set of two matrixes." (Joyner et al., 2015)

The authors also defines IPR as:

> "the difference in percentage of the number of dark-colored pixels present at the same coordinates with respect to the total number of dark-colored pixels in both cells for a given set of contiguous pixels." (Joyner et al., 2015)

Joyner's agent tests the DPR and IPR across rows and columns to see if the addition of the answer image closely relates to the known DPR's or IPR's. Agent

3 ultimately uses a voting system to accumulate information from both measures and multiple tests.

1.2 Preprocessing

Before any calculations take place, images need to be in PIL's '1' mode. This converts the image to a series of boolean values representing white and black.

1.3 DPR

This calculation leverages the conversion to the boolean values. Each numpy element representing a black pixel is stored as True. The function get_dpr(), sums the trues to get the number of black pixels in each then divides this by the total number of pixels.

1.4 IPR

The function get_ipr() finds the ipr given two matrices. First, the function calculates the number of intersecting black pixels as the sum of True values in the Logical Or combination of the two matrices. The IPR then is the intersecting black pixels divided by the number of black pixels in either matrix.

1.5 Testing Logic

In a 2x2 matrix approach, the agent would only need to calculate the DPR and IPR for the cells A to B, and A to C representing the row and column slices of the matrix as shown in Figure 1. To select an answer, the agent could then calculate what the DPR and IPR would be for addition of each answer image to either C (row) or B (column). The answer image with the closest DPR and IPR receives a vote. After all calculations are done, the answer image with the highest votes from DPR or IPR is the selected answer.

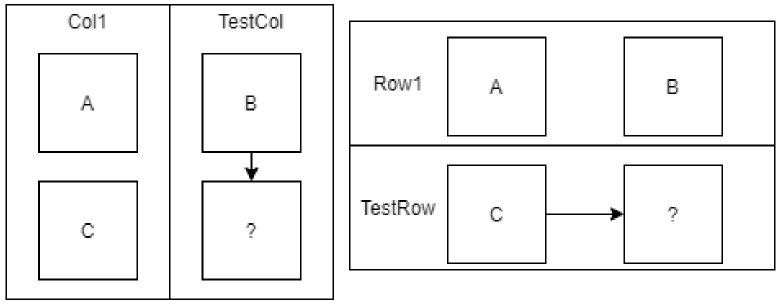

Figure 1— Visual of matrix row and col tests.

However, for a 3x3 matrix, further abstraction is necessary to apply this methodology. The agent uses the 2x2 approach as previously described for the basis of the analysis except it uses composite cells which are the logical AND of two images. For instance, the agent incorporates the cells D & E, and B & E to calculate the DPR and IPR for (D&E) to F, and (B&E) to H as shown in Figure 2. These composite images incorporate more information than if only E were used for the calculations.

Additionally, the agent employs an aggregate change in DPR and IPR for row and column levels. This encapsulates more information from the larger matrix. Some matrices may have information embedded in the series of rows or columns. Therefore the agent attempts to capture this information by row and column. For example, in Figure 2, the difference in the Change in Row % and the Observed Change in Row % from a given answer image should be relatively similar. In practice, the agent operates on this principle with a tolerance of 0.15 in either DPR or IPR.

The agent also calculates the DPR and IPR for the left diagonal A to E. Then the agent tests if the percentages are similar to E to each answer image.

Ultimately, if the calculated DPR and IPR is similar to the tested column, row or diagonal relationship then the agent votes for that tested answer cell.

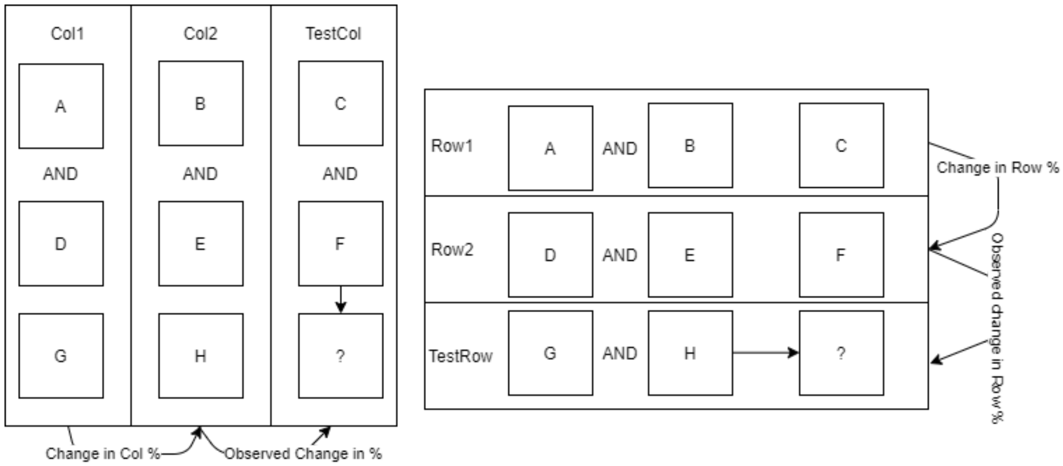

Figure 2— Visual of matrix column and matrix row aggregate change.

2 PERFORMANCE

The agent correctly answers 9 of 12 Basic Problems C and 8 of 12 Test problems. The proceeding sections discuss the struggles, potential improvements, and efficiency of the agent.

2.1 Struggles

The agent struggles with diagonal relationships where there is either increasing complexity or reflections. The Basic Problem C-07 in Figure 3 exemplifies where the agent struggles the most. Here the agent attempts to find the answer image that fits the most similar DPR and IPR to the row or column level information. However, this doesn't make sense as the change in DPR and IPR is the opposite sign. The true answer appears to either be a reflection around E or a reflection on the second row. This issue persists into the problem sets D and E. The agent needs to address the more affine transformations.

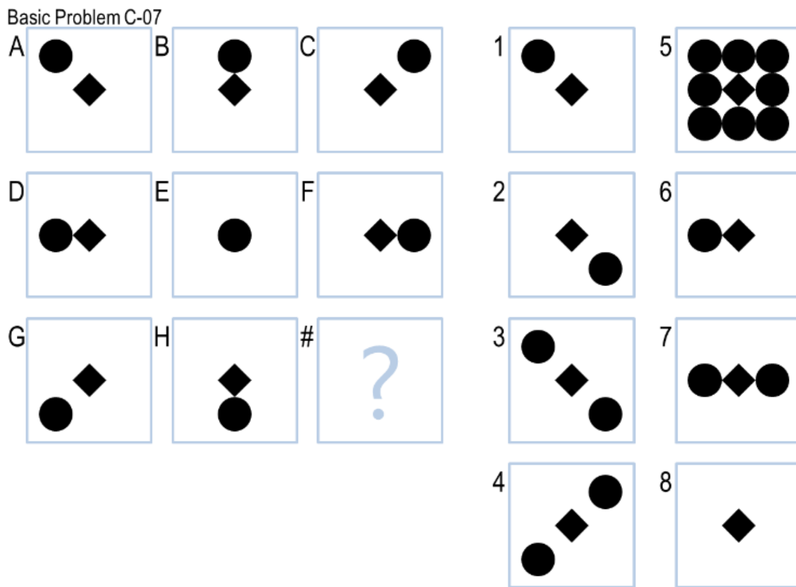

Figure 3— Basic Problem C-07 example.

2.2 Improvements

To address the failure to recognize the reflection relationships, the agent could incorporate information from the right diagonal. In this situation, the test would be to compare DPR and IPR from (A&E) to answer set to the diagonal (C&E) to G.

2.3 Efficiency

The agent performs fairly quickly. The Agent.Solve method took 1.855 seconds in total while being called 96 times. The agent far out performs the agent in Milestone 2 which used more traditional image processing comparisons.

3 Potential Feedback

How can I improve the logic of my tests to better address reflection? Should I also be testing DPR/IPR with combining frames based on OR or XOR?

4 References

1. Joyner, D., Bedwell, D., Graham, C., Lemmon, W., Martínez, Ó., & Goel, A.K. (2015). Using Human Computation to Acquire Novel Methods for Addressing Visual Analogy Problems on Intelligence Tests. ICCC.