RPM Milestone Final Journal

Athanasios Grivakis | agrivakis3@gatech.edu

High-level Functioning of the Agent

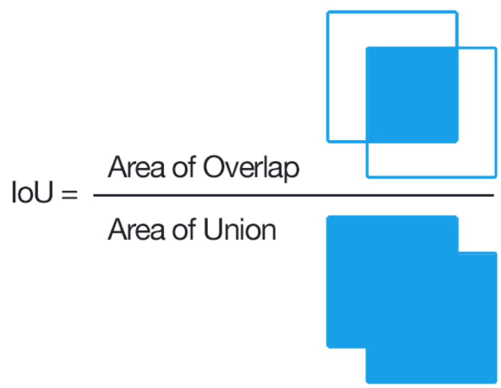

Fig. 1: Geometric visualization of intersection pixel ratio

The agent solves Raven's Progressive Matrices on a high level by performing several steps to obtain the answer which best fits the problem. The first step is utilizing a variety of metrics to evaluate similarity among given matrices within the problem (for example along a row, along a column, or along a diagonal), to gain a sense of the transformation that describes the difference between those initial matrices. The transformation is encapsulated by the difference between the images using a variety of metrics. Next, the agent takes the evaluated transformation and applies it to a matrix in the problem which is "next" to the answer choice. For example, in a 2x2 matrix problem, the agent will apply the chosen row transformation and column transformation to the matrices next to the answer choice matrix. In a 3x3 problem, the agent might consider the matrices along the diagonal(s) in addition to rows and column relationships. Now, the agent will perform the same transformation among matrices adjacent to the answer matrix and the various answer choices. While it performs the transformation, it will take the difference between this applied transformation and the result of initially applying the transformation earlier. The agent keeps track of how similar each transformation applied to the answer choices is to the transformation applied to the initial matrices, assigning high similarities with high scores and low similarities with low scores. Finally, the agent selects the answer choice which is associated with the highest calculated score.

I will elaborate a bit on the metrics used by the agent, in order for the reader to gain a better understanding of what kind of metrics the agent is using to measure similarity between images and thereby encode such information as a type of translation that can be performed between images. These metrics include dark pixel ratio, intersection pixel ratio, and bitwise operations. Dark pixel ratios essentially measure the ratio of dark pixels to the entire image; this is useful in giving the agent a sense of location for the shapes within any given progressive matrix. Intersection pixel ratios measure the ratio of the dark pixels in a combined image to the ratio of dark pixels in the union of two images. This is analogous to a ratio of the logical intersection to the logical union. This is useful for gaining a sense of "change" among separate shapes in two different images. The agent utilizes these two metrics to gain a sense of change among two different matrices' shapes by recording differences in the pixels; this information encapsulates what is necessary to encode a "transformation" for the agent. The agent also uses bitwise operations in some circumstances; these include "or", "and", "xor", and "negation of xor". These operations encode similarity between two input images and can be treated as transformations by the agent when applied among consecutive images in a row, column, or diagonal. This is how the agent calculates and encodes transformations on a high level. These metrics can be computed between images once they have been binarized, which OpenCV provides functionality to obtain.

Agent Performance Across Raven's Progressive Matrices Problems

I will define performance as adherence to two principles: consistency and generalization. Consistency refers to how often the agent performs correctly among problems of a similar type, and generalization refers to how well the AI approach used by the agent could be scaled and further applied to varied cases. For the Basic problems, the agent correctly solved 35 problems and missed 13 problems. For the Challenge problems, 14 were solved correctly and 34 were missed. For the Test problems, 30 were passed and 18 were missed. For Raven's problems, 30 were passed and 18 were missed. The agent performed in about 2.87 seconds. For a more specific breakdown, within Basic B, 7/12 problems were passed. Within Basic C, 10/12 problems were passed. Within Basic D, 9/12 problems were passed. Within Basic E, 9/12 problems were passed. This shows that, on average, 75-80% of the problems within the Basic problem set were solved for each set, which displays consistency. Within Challenge B, 5/12 problems were passed. Within Challenge C, 3/12 problems were passed. Within Challenge D, 1/12 problems were passed. Within Challenge E, 5/12 problems were passed. This shows that, on average, 25-30% of the problems within the Challenge problem set were passed. Compared to the Basic problems, this is somewhat of a low consistency. Within Raven's B, 8/12 problems were passed. Within Raven's C, 7/12 problems were passed. Within Raven's D, 5/12 problems were passed. Within Raven's E, 9/12 problems were passed. This shows that, on average, 55-65% of the problems within Raven's problem set were passed. This is more consistent than the Challenge set but less consistent than the Basic set. Within Test B, 6/12 problems were passed. Within Test C, 8/12 problems were passed. Within Test D, 7/12 problems were passed. Within Test E, 9/12 problems were passed. This shows that, on average, 60-70% of the problems within the Test problem set were passed. This is less consistent than the Basic set but more consistent than the Challenge and Raven's sets. Within an overall consistency of around 70%, I would say that the performance of the agent is consistent. Below is a table summarizing the above statistics:

	B	C	D	E	B	C	D	E	B	C	D	E	B	C	D	E
Pass	7	10	9	9	5	3	1	5	8	7	5	9	6	8	7	9
Miss	5	2	3	3	7	9	11	7	4	5	7	3	6	12	5	3

Fig 2: Display of agent performance (green - basic, red - challenge, blue - Raven's, yellow - test)

For generalization, the agent uses a type of means-ends analysis approach, which involves taking an action that reduces the difference between an initial and final state. This is done using the dark and intersection pixel ratios, and bitwise operations, then evaluating the differences among the row, column, and diagonal transformations among different pairings of matrices. This approach can easily be generalized to solve more comp[ex problems, and is scalable; one only has to change the arguments from matrices of images to general things to be compared, and the same means-ends analysis still holds.

Good Performance and AI Reasoning

The agent solves Basic B-02 by computing a row transformation among matrices A and B (using intersection pixel ratio and dark pixel ratio) and then computes the difference between C/answer and A/B. It correctly chooses the answer which results in the least difference between row comparisons. The agent does a similar approach for Basic B-04, as well as Basic C-01. The agent likely does well on these problems since the approach is straightforward, and is applicable to all these problems in the same manner.

Struggles and AI Reasoning

The agent attempts to solve Challenge D-03 by computing row transformations among matrices in the first two rows, as well as the first two matrices in the last row (using intersection pixel ratio and dark pixel ratio). It also computes column and diagonal transformations in a similar manner. It then takes the row transformation between H and the answer, the column transformation between F and the answer, and the diagonal transformation between E and the answer, and seeks the choice that produces the minimal difference among those transformations and the initially computed transformations from matrices A through H. It incorrectly chooses the answer which results from the least difference between comparisons. The agent does a similar approach for Challenge D-05. The agent likely does not do well on these problems since the images involved are more complex and therefore, the transformations used to result from the intersection pixel ratio and dark pixel ratios may have been too much of an estimation to be completely accurate, and the agent selected the wrong answer. Additionally, it is likely that the agent ran into inaccuracies in calculating the diagonals since this necessarily involved more steps, and leaves more room for the agent to make errors.

Approach Identification

The agent uses an approach that is consistent with means-ends analysis. Means-ends analysis involves taking an action that reduces a computed difference among an initial state and a goal state in order to reach an answer. This can possibly occur once, or over many iterations in order to reach the desired outcome. The agent uses a means-ends approach by treating the initial state as the initial row/column/diagonal transformation (calculated using dark pixel ratio and intersection pixel ratio), the goal state as the row/column/diagonal transformation between the answer choice(s) and an adjacent matrix, and calculating the "difference" between both sets of transformations, to find the minimal difference to obtain the answer. The difference is calculated using similarity metrics based on the dark pixels within both sets of transformations among images. In this way, the agent uses the initial = state, goal state, and difference to calculate and obtain the answer.

Agent-Human Comparison

The agent performs the problem somewhat differently than the approach that a human would take in solving the problem. A human will likely take two images (in a given row, column, or diagonal), and then make a mental representation that encapsulates the transformation between those two images. The person will then look at the images adjacent to the answer choice matrix, and then determine what the answer should look like based on applying the same change/transformation to those adjacent matrices. They will then look at the answers and determine which answer choice most closely resembles that answer choice. If it is an exact match, they will choose that answer; otherwise, the person will usually consider if there are more relationships that can be derived among the image, such as by considering all four matrices as a block and looking for symmetry, or considering more transformational relationships. The agent, in contrast, obtains a calculated representation of a transformation using specified metrics and then applies that transformation on the adjacent matrices to the answer choices and the answer choices themselves. It then chooses as the answer the choice which minimizes the difference between the two transformations. So, rather than apply the transformation and then see which answer fits, it considers the difference between the two sets of transformations and determines which answer choice provides a minimal difference. To a human, the agent's approach might seem unintuitive and calculated. The agent's approach is generally more consistent, but might be more inaccurate due to its specific method.