Project 3: Raven's Progressive Matrices Journal

Sergiy Palguyev

sergiy.palguyev@gatech.edu

Introduction

The third iteration of Raven's Progressive Matrices solving algorithm began with the solution from Project 2 as the base line. Project 2 primarily focused on comparing the difference in black pixel ratios between the different images in order to derive some order and infer a possible solution. For project 3, the strategy devised is to observe and determine patterns that follow from either the pixel ratios or pixel count which may lead to the correct answer.

Strategy

The patterns explored will comprise evaluation the increase in black/white ratios across rows and columns, determining whether each progressive layer of the problem set is a composite of the layers before it, observing whether the images overlay one another in subsequent rows or columns, and a comprehensive analysis of whether images have common sections which may clue to the commonality of a possible answer.

Additionally, an educated guess algorithm will ensure that no problem is skipped without at least trying to determine the answer based on limiting the possible amount of choices.

Implementation

In order to implement these features both the image matrix representation sand their ratios will have to be evaluated for different features and compared. The code that is implemented will follow previously established algorithms in order to minimize the number of problems to work and focus on. It is evident that the order of analytical operations can influence the outcome of the results.

The algorithm should be implemented in such a way that the best answer from the possible solving techniques will be used as the final answer to a problem. This will ensure that each evaluation is looked and judged fairly by the algorithm.

1st Submission

When was the submission sent in?

2019-04-18 07:28:24 UTC

What did you change for this version? Why?

For this check-in, the latest Project 2 code was checked in. This code determined a base line against which all future code will be measured. This way, the progress and success of the algorithm and its features can be determined.

How does this algorithm compare to human thought?

This algorithm compares well to human thought processes. The simplest human observation is determining patterns of increasing or decreasing weight. If one determines that some item in an image is getting larger, or smaller, one can deduce that the final image under question will also get larger or smaller based on the analyses of the problem set. As such, observing the black to white ratios change across the problem matrix provides a rudimentary human cognitive problem solving skill of increasing/decreasing weight of the image.

How is the performance?

Data Set	Basic D	Basic E	Test D	Test E
Correct	3	3	3	1

Table 1: 1st Submission Results

As observed in Table 1, the performance of the algorithm, specifically in section D and E were 25% accuracy in Basic problem set, for both, and 25% for set D and 8% accuracy for set E in the Test problem set. The algorithm, so far, is only good at solving some solid-fill problems or problems with increase in shape complexity.

2nd Submission

When was the submission sent in?

2019-04-20 03:48:32 UTC

What did you change for this version? Why?

For this check-in, image overlay analysis is added. Each row and column is analyzed for image overlay properties. If such a property is determined, the row (G and H) or column (C and F) images that lead to the problem image are analyzed for overlay with all possible answer images.

How does this algorithm compare to human thought?

This algorithm was the logical next step in development. As human thought would deduce that some images are an overlay of the previous images. A human would be able to determine from the 3rd images in a row or column that the image is an overlay of the previous two. A human would then be able to pick an image form the Answer set that would resemble the same overlay from the last row or column.

How is the performance?

Data Set	Basic D	Basic E	Test D	Test E
Correct	3	6	4	7

Table 2: 2nd Submission Results

As observed in Table 2, the performance increased dramatically for Basic and Test E test set. Basic and Test sets D had almost no effect on them. Overall the accuracy of the algorithm has increased to 25% for Basic D, 50% for Basic E, 33% for Test D and 58% from Test E.

3rd Submission

When was the submission sent in?

2019-04-21 09:51:05 UTC

What did you change for this version? Why?

For this check-in, composite image analysis was implemented into the overall algorithm. This analysis is designed to evaluate the problem set for image features that show combinational patterns. The images are evaluated across rows and columns and if the pattern is determined, the last row (G and H) or column (C and F) are then matched with possible answers that may exhibit this behavior as well.

How does this algorithm compare to human thought?

This algorithm compares to the human thought process similarly as in the previous submission. Whether comparing the common areas of an image or combining multiple images together, humans easily can observe such patterns with very little skill. The observation of common features across rows or columns is a simple form of image processing and analysis performed by humans daily.

It is often noted that humans can see simple shapes in complex structures (i.e. cloud formations, face recognition, etc). It is this skill which is being explored by overlapping and combining images to observe either common areas or combined resulting images in order to find a matching answer. This is a fairly basic skill since no recognition of object type is being done, simply pattern matching is being performed to a very rudimentary level.

How is the performance?

Data Set	Basic D	Basic E	Test D	Test E
Correct	3	7	4	7

Table 3: 3rd Submission Results

As observed in Table 3, the performance increased very slightly from the previous submission. Overall the accuracy of the algorithm has increased to 25% for Basic D, 58% for Basic E, 33% for Test D and 58% from Test E.

4th Submission (2019-04-21 17:23:41 UTC)

When was the submission sent in?

2019-04-21 17:23:41 UTC

What did you change for this version? Why?

For this check-in, the algorithm implemented determined whether the third row or column was the summation of the previous two rows or columns. This algorithm is instrumental in solving problem sets where the addition of features is a common characteristic. For this behavior, observing the feature set allows to predict the further addition of pixels as a feature of the answer set.

How does this algorithm compare to human thought?

This algorithm relates to human thought as humans tend to extrapolate that the tendency for features to be added to an image will continue as a characteristic for the unknown image. Humans tend to notice features that were added or removed from an image set and continue the pattern into the possible answers. As such this algorithm should closely mimic human evaluation of such a problem. This behavior is somewhat harder than other since features may become irregular or small, making distinct details difficult to determine, for a human.

How is the performance?

Data Set	Basic D	Basic E	Test D	Test E
Correct	4	10	4	8

Table 4: 4^{th} Submission Results

As observed in Table 4 the results accuracy increased dramatically. Both Basic and Test D problem sets are solved with an accuracy of 33%. Basic E now attains an accuracy of 83% and Test E an accuracy of 66%, showing a fairly modest increase in solve rates from the previous implementations.

5^{th}-7^{th} Submissions

When was the submission sent in?

5^{th} submission 2019-04-22 03:25:44 UTC
6^{th} submission 2019-04-22 05:15:23 UTC (re-submission of same code)
7^{th} submission 2019-04-22 05:18:11 UTC (re-submission of same code)

What did you change for this version? Why?

For this check-in, the changes comprised an edge analysis of the G and H images or the C and F images to match the pixel ratios to the answers in between a set of thresholds. The upper bound tested to produce good results was chosen as 0.05 and the lower bound was chosen as 0.02. These bounds were attained by trial and error to produce good results for the Basic problem set.

Additionally, if none of the previous algorithms produced viable results, the end of the solution evaluation would attempt to make an educated guess. This guess would first throw out any images that were too similar to the ones from the problem set and narrow down the amount of possible solutions as much as possible. Subsequently, a random value was chosen from the remaining possibilities.

In order to exercise and observe the effect of guessing, the same code was submitted 3 times, to monitor its effects on the possible answer outcomes. Ass seen in the table below, the effects are minimal in changing how well the algorithm works.

How does this algorithm compare to human thought?

These algorithm additions translate well into human thought as people would often go to the edges of the problem matrix to try a shortcut to the solution instead of evaluating all 8 problem images. If a solution could be determined quickly form the 4 images at the edges, then a human agent would say the solving time was optimized for speed.

Additionally, if not penalized a human agent would always guess an answer rather than skipping the question altogether. The guess must still be an educated guess and thus the algorithm first filters out some answers which definitely would not fit.

How is the performance?

Data Set	Basic D	Basic E	Test D	Test E
5th Correct	8	11	7	8
6th Correct	8	12	7	8
7th Correct	8	11	7	9

Table 5: 5th, 6th, and 7th Submission Results

As observed in Table 5 the results again undergo a dramatic increase in accuracy. For each submission, Basic D has an accuracy of 66%, Basic E has an accuracy over 90%, Test D has an accuracy of 58% and Test E has an accuracy over 66%. As demonstrated, the educated guess does not add much variability and only serves to increase the chances of getting more correct solutions rather than decrease the correct solution count.

Conclusion

In conclusion, the final agent has undergone a dramatic accuracy increase. Although not perfect, the algorithm is successful in determining more than 58% of answers for every data set passed to it, with some datasets solving for 92% of the problems. Although some random guessing is implemented if no method can solve a problem, the guesswork is reduced to a subset of answers most likely to contain the correct one.

In summary, the accuracy improvement is shown in the table below.

Submission	Basic D	Basic E	Test D	Test E
1st Accuracy	25%	25%	25%	8%
2nd Accuracy	25%	50%	33%	58%
3rd Accuracy	25%	58%	33%	58%
4th Accuracy	33%	83%	33%	66%
5th Accuracy	66%	92%	33%	66%
6th Accuracy	66%	100%	58%	66%
7th Accuracy	66%	92%	58%	75%

Table 6: Accuracy summary table

How would you characterize the overall process of agent design?

The overall process of agent design was mostly deliberate. Certain features of the agent which did not make it into submission were solution methods which did not improve accuracy or even became detrimental to the overall problem solving. Generally, there are sets of problems which were presented and each set was boiled down to components and changing variables. Those variables were then attacked one by one in order to attempt a solution method.

Trial-and-error, however, was also an integral method to creating the algorithms. More specifically, the thresholds by which certain characteristics of the images were determined to be present had to be determined through rigorous trial and error. Certain features had wide margins between passing and failing cases, which made the thresholds easy to determine. Other features made the job much more difficult and the thresholds were determined by running multiple sessions of the algorithm and picking the threshold which produced the most correct answers.

How similar do you feel your final agent is to a human approach?

The agent is very similar to human approach to image problem solving. Utilizing methods such as image overlays, additions, subtractions, common features, shortcuts across certain rows and columns as well as random guessing all being methods employed by human minds to solve the same types of problems.

What improvement would you make?

Given more time and resources, I would attempt to add weights to each answer. Every method would run through the same problem set and attach a weight to the most likely answer. At the end, all weights would be tallied and the one with the most weights would be the answer chosen.

Additionally, given the proper amount of time, shape determination or categorization would also be a feature to implement, since with recognition of certain shape features, determining the correct answer would become much more promising, as the answer set could be sorted by containing the proper shape.

Finally, a generate and test would be an additional feature which would generate possible answers based on the problem images and the features which could be derives from executing the previously mentioned methods. Instead of determining or comparing the answer and problem images to come predefined threshold, a new set of possible answer images would be generated and each compared against the provided possible answer set.