Project 1: Raven's Progressive Matrices

Marc Micatka
mmicatka3@gatech.edu

1 INTRODUCTION

Raven's Progressive Matrices (RPM) are visual analogy tests used to measure abstract reasoning and fluid intelligence. Originally developed by John Raven in 1936, the test consists of 60 non-verbal pattern-matching and visual analogy questions. 2x2 or 3x3 image matrices are presented with one item missing. The task, as shown in **Figure 1**, is to select the visually analogous answer that best completes the matrix.

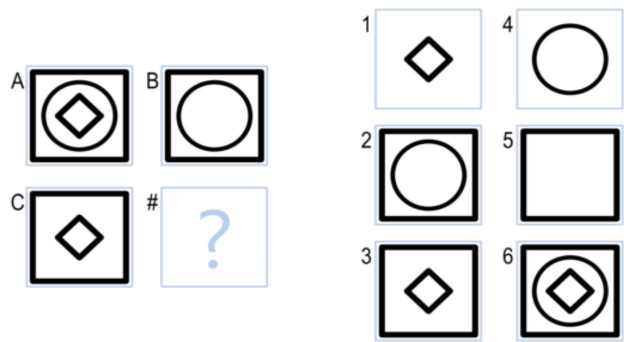

Figure 1 *- Raven's Matrix for a simple 2x2 problem.*

In Project 1, the problems are 2x2 and the transformation is often basic – a simple geometric transformation of the original image. After the images are loaded, the algorithm computes various transformations between A and B and between A and C, giving each transformation a similarity score.

Table 1 *– Transformations used in Project 1.*

Transformations	Description	Weight
Identity	Compare A and B directly	1.0
Reflection: X	Reflect B across X-axis	0.50
Reflection: Y	Reflect B across Y-Axis	0.50
Rotation: 90°	Rotate B 90° clockwise	0.25
Rotation: 180°	Rotate B 180° clockwise	0.25
Histogram Comparison	Calculate the ratio of white/black pixels	0.25

The algorithm calculates all metrics in **Table 1** for each suggested answer. It then compares these proposed matrices with the base transformation matrices T_h and T_v and the best match is then chosen from the possible answers.

2 JOURNAL ENTRIES

2.1.1 *Submission 1: Friday, 6 September 2019, 18:30 UTC*

Problem Set	Correct	Incorrect	Results
Basic Problems B	7	5	58.3%
Test Problems B	8	4	66.6%
Challenge Problems B	6	6	50%
Runtime	-	-	5.96877 s

What did you change for this version? Why?

This was the first submission. No revisions were made to this version. The first submission used a very simplistic version of the algorithm described in the **Introduction**. It only considered the horizontal transform, T_h, and it only used identity, rotation, and mirrored transformations (no pixel counts and no pixel weights).

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it "think" similarly to how you think, or differently?

In the most basic sense, this algorithm does attempt to mirror human cognition. It looks for similarities to determine the transformation T. The first version is only searching for the basic transformations – rotation and mirroring about the x- and y-axis. It does not perform any higher-level pattern recognition or logical operations (searching for image subtraction or addition).

In addition, humans are performing multiple layers of analysis, perhaps without recognizing what they are doing. We first search for the obvious transformation (such as the transformations this algorithm looks for). If that doesn't yield a satisfactory answer, we look for more complex patterns. This second level is both more abstract and more computationally intense. This agent stops after the first level of analysis and does not attempt to look for more complex patterns.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

For the first attempt, the agent performed admirably. It passed the 7/12 performance threshold. The agent struggled with basic and challenge problems that require the use of a vertical transform, such as Basic Problem B-05 (see **Figure 2**). This problem is difficult to solve horizontally, as the agent would need to recognize the transformation from a triangle to a square. It's quite simple to solve using a vertical transformation (from A to C) because it's mirrored over the y-axis.

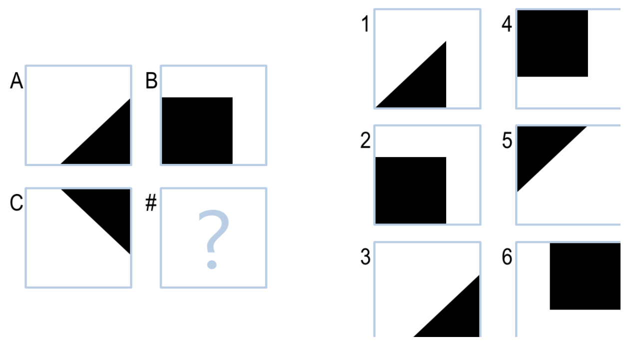

Figure 2 *- Simple mirrored transformation.*

The efficiency of this algorithm is bounded by its transformation checks between adjacent neighbors which yields an efficiency of $T(n) = O(n^2)$.

2.2 Submission 2: Tuesday, 10 September 2019, 19:35 UTC

Problem Set	Correct	Incorrect	Results
Basic Problems B	8	4	67% ▲
Test Problems B	9	3	75% ▲
Challenge Problems B	6	6	50% ▶
Runtime	-	-	5.9857 s

What did you change for this version? Why?

For the second revision, I kept the same similarity measures (identity, refection, and rotation) but I included a simple check to identify which transformation – vertical or horizontal – returned the best similarity score. From that check, the solution was chosen. This small change improved the performance on the basic problems and the test problems by one each, a relatively small improvement.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it "think" similarly to how you think, or differently?

This version of the algorithm does a better job of mirroring human cognition by including vertical transformations. The consideration of vertical and horizontal transformations is similar to how humans approach the problem. The scoring metric is also similar in concept to what people do but different in execution. Humans are not computing a decimal metric to determine how similar objects are. But we do recognize the similarities between two shapes offset by a simple affine transformation.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

After assessing the performance, I'm still not successfully catching simple transforms, like the one shown in **Figure 2**. Although I'm checking for vertical transforms, if the vertical transform does not score high enough, it will not be selected. Because of how I am calculating similarity (pixel-to-pixel comparison) it is not robust to the type of shape transformations that are occurring in B-05. This is because the similarity of a triangle compared with the same triangle rotated 90° is not the same similarity score as a square compared with a rotated square. Even if the transformation is identical, some similarity scores will be off. The efficiency is unchanged from submission 1, the runtime has increased marginally.

2.3 Submission 3: Saturday, 14 September 2019, 01:05 UTC

Problem Set	Correct	Incorrect	Results
Basic Problems B	10	2	83% ▲
Test Problems B	9	3	75% ▶
Challenge Problems B	7	5	58% ▲
Runtime	-	-	6.55129 s

What did you change for this version? Why?

For the third revision, I added an additional metric to my feature matrix – the white pixel ratio difference between the two images. As suggested in *Using Human Computation to Acquire Novel Methods for Addressing Visual Analogy Problems* *on Intelligence Tests* (Joyner, 2015), the ratio between white/dark pixels in an image should be roughly consistent across transformations. This helps solve problems like Basic Problem B-12 with no understanding of shapes or implementation of logical operations between images.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it "think" similarly to how you think, or differently?

This revision doesn't fundamentally change the operation of the classifier. Again, the approach is similar to human cognition, but the metrics used to achieve the solutions are non-human. The introduction of pixel count is not a typical human though process, but we do have human analogs (like looking at color fill – shaded vs empty vs full).

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

This submission improved the performance of the basic and challenge problems by 3 overall, a decent improvement. The solver is not answering basic questions 4 and 9 correctly as well as missing challenge questions 1,3,4,8, and 10. 4 is a fairly confusing question addressed in submission 4, 9 is another confusing question that deals with image fill that I don't have a great solution for. The efficiency is unchanged from previous revisions, the runtime is marginally increased in this version.

2.4 Submission 4: Sunday, 15 September 2019, 21:25 UTC

Problem Set	Correct	Incorrect	Results
Basic Problems B	11	1	92% ▲
Test Problems B	8	4	66% ▼
Challenge Problems B	8	4	66% ▲
Runtime	-	-	6.06204 s

What did you change for this version? Why?

To solve some of the more vexing RPMs that my agent was still missing, such as Basic B04, I updated how I calculate my similarity matrix with a weighted distance calculate. If there is an identity match between transformations, that is of greater interest to me than a histogram match, or a reflection match. By playing with the weights, I was able to achieve a better score on my Basic and Challenge problems, but I started missing one question of the Test set.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it "think" similarly to how you think, or differently?

The weighted approach adds a bit more cognition to the calculations. When humans approach the problem, we value simpler, more direct transformations more than complex ones. This algorithm attempts to reflect that type of thought process.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

This version performs nearly perfectly on the basic questions (misses B-09) and misses four challenge questions (1, 3, 4, and 8). Challenge questions 1 and 3 are simple rotation operations that can be solved by adjusting my weight parameters or improving my recognition of perfect transformations. Questions 4 and 8 are more challenging logic operations that I do have a great answer for yet. The efficiency is unchanged, the runtime has slightly improved over the last iteration.

2.5 Submission 5: Sunday, 15 September 2019, 21:25 UTC

Problem Set	Correct	Incorrect	Results
Basic Problems B	12	0	100% ▲
Test Problems B	8	4	66% ▶
Challenge Problems B	8	4	66% ▶
Runtime	-	-	6.51587 s

What did you change for this version? Why?

For this revision, I added a check for perfect transformations and updated my weighting algorithm. If a perfect geometric transformation was found, the program would search for a perfect match among the proposed answers. If no match was found, it would default to the regular search algorithm. This, along with the updated weighting strategy, nabbed the one basic question I was unable to solve.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it "think" similarly to how you think, or differently?

See above answers. There are only minor revisions to this submission, and it doesn't fundamentally alter the answers to these questions.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

Achieved a perfect score on the basic questions. Still only achieving 8 out of 12 on the challenge and test questions. I continue to struggle with the subtraction operations and questions involving "fill" even with the introduction of a pixel count metric. Its efficiency is unchanged.

3 CONCLUSION

How would you characterize the overall process of designing your agent? Trial-and-error? Deliberate improvement? Targeting one type of problem at a time?

I started with a very basic approach that was based on some of the fundamentals of solving RPMs as laid out in *Reasoning on the Raven's Advanced Progressive Matrices Test with Iconic Visual Representations* (Kunda, 2012) The result of this first attempt was submission 1, which correctly solved 7 of the 12 basic problems and 8 of the 12 test problems. From there, I used a combination of a deliberate approach rooted in strategies outlined by various publications and individual problem targeting to solve specific problems that seemed trivial.

How similar do you feel your final agent is to how you, a human, would approach the test? Why or why not?

On a (very) high level, the reasoning of my agent behaves similarly to a human in that it first attempts the solution using very fast but simplistic reasoning. If the first layer or attempt fails, it will try more computationally expensive and abstract methods.

In both humans and my agent, this includes first searching for simple vertical and horizontal transforms like rotation and mirroring as well as simple logical operators like addition and subtraction of images. If this fails to give an adequate solution, humans will move on to try more complicated methods involving pattern recognition.

The agent will also try more complicated methods, but they will involve non-human behaviors like measuring pixel ratios.

The major differences, and shortcomings, between my artificial agent and human cognition are in pattern recognition and feature identification. Humans solve the more complicated Raven's Progressive Matrices by recognizing shapes, transformations, and patterns. This method can make some problems, as shown in Basic B9, very simple for humans but very difficult for algorithms that lack the ability to count edges or recognize shapes.

What improvements would you make if you had more time and/or more computational resources?

With OpenCV, it would be much simpler to identify shapes, key features like corners and intersections, and transformations using connected components, blob analysis, and shape detection. This would make problems like Challenge B-09 easier for an artificial agent to solve.

Using the allowed libraries, I would like to expand my solver to accommodate 3x3 images by assessing the various analogies that exist in a 3x3 compared to a basic 2x2. In a 2x2 analogy, we have two main relationships – vertical and horizontal. In a 3x3 we also include the diagonal relationship and we must contend with multiple vertical/horizontal transformations. In addition to *finding* all these relationships, I must also make the decision on how to weight the various transformations to pick a final solution.

REFERENCES

Joyner, D. B. (2015). Using Human Computation to Acquire Novel Methods for Addressing Visual Analogy Problems on Intelligence Tests. *Proceedings of the Sixth International Conference on Computational Creativity.* Provo, Utah.

Kunda, M. M. (2012). Reasoning on the Raven's advanced progressive matrices test with iconic visual representations. *34th Annual Conference of the Cognitive Science Society.* Sapporo, Japan.