1. Introduction:
   1. Self Introduction
   2. Slack Channel
   3. Introducing the topic
2. Describe Raven’s and Example Problems
3. Overview
4. Similarity Metrics
   1. Jaccard
   2. Alignment
   3. Similarity Procedures
5. Transformations
   1. Two dimensions, four classes
   2. Unary Affine Transformations: rectilinear rotations and reflections
   3. Unary Set Transformations:
      1. Columns
      2. Given parameters A and B, Transformation T transforms C into the thing in the last column.
      3. For example, add\_diff
   4. Binary Set Operations:
      1. Columns
      2. Given parameter C, Transformation T transforms A and B into the output image.
      3. For example, inv\_unite
6. Analogies
   1. Also two dimensions, and four classes
   2. Simple pair analogies
   3. Simple 3-tuple analogies
   4. Recursive 3-tuple analogies:
      1. Composed of two or more simple analogies
      2. Each simple analogy represents a sub-problem, for example…
   5. Recursive pair analogies:
      1. The same as the recursive 3-tuple, but as the size of basic element is smaller, we can include more sub-problems in a recursive analogies here.
      2. The purpose of recursive analogies is to include more information in our computation for the solution.
7. More Analogies
   1. Expand the matrix by copying the entries in a cyclical way
   2. So that the original spatial relation is preserved locally everywhere in the expanded matrix.
   3. For example, in the original matrix, B is on the left of A, in the expanded matrix, B is always on the left of A everywhere.
   4. Encompass new matrices with quadrilateral of different shapes. The intuition of this is different shapes of quadrilateral corresponds to different new spatial relations in the new encompassed matrices.
   5. Do the same thing for 3x3 matrices.
   6. However, we do expect that human’s do the same things. The goal of this method is to give us a formal and automatic way to generate all the analogies we need to solve the SPM.
8. Integration Strategies
   1. Raise the question.
   2. Describe the 2x2 problem
   3. Briefly, describe the 3x3 problem
   4. Heuristics
   5. Given the heuristics and functional dependencies, we have two dimensions, and 6 strategies in total.
   6. Describe them one by one in terms of optimization order.
9. Strategy Comparison
   1. Describe the structure of SPM test.
   2. On the left, M-strategies, optimizing the MATO score, almost the same performance.
   3. On the right, O-strategies, optimizing the O score,
      1. O-confident has the same performance as M-confident on the left.
      2. The other two contrast sharply with their counterparts on the left. The less they rely on the heuristic, the worse performance they have.
      3. It is mainly because the trap option … fully understand the matrix part.
   4. Bubble plot:
      1. Detailed information about how each problem is solved by each strategy.
      2. Encoding, size and darkness, and correctness.
      3. On the top, strategies may arrive at the same correct answer by via different path, otherwise …
      4. At the bottom, some bubbles, representing very low MAT scores, are two small to show according to the scale.
   5. Scatterplot:
      1. Bubbles to datapoint, to see the general distribution.
      2. X axis and Y axis
      3. On the left, M-strategies. Most datapoints are around or below the diagonal.
      4. On the right, O-strategies, with worse performance, more datapoints are far above the diagonal.
      5. Hypothesis.
      6. Address the reviewer’s comment: not interesting?

The perspective from which we consider the problem is different. Our goal, at least one of our goals, is to develop AI agents to solve different intelligence tests. We treat our agents as outsider of this area, without no prior knowledge of the Raven’s test. And imagine, that if we give these data to an outsider and it can derive a useful strategy to this test from these data, won’t it be a very interesting process. But we do admit that if we treat our agents as experts in this area, like most of us, a lot of prior knowledge has been “hardcoded” in our brain when we are considering these test problem, in this case, it is indeed not an interesting hypothesis.

1. Besides strategies, we see how transformations and analogies affect the result.
   1. Grouping
   2. Describe encoding
   3. 1-to-1
      1. Strength and weakness
      2. For example, S + diff and, Set operation/R.
   4. Incremental way:
      1. Monotonically increase in both horizontally and vertically.
      2. But the vertical increases are much more obvious.
      3. Address the reviewer’s comment: Can we get rid of the analogy groups, and study only transformation groups?

No, we can’t. See the 1-to-1 plot. S is more powerful than HVR, but we start with S in the incremental plot. That’s is why the variation is in the horizontal direction is not so obvious…

If we define the analogy groups at a finer level, we will probably see more increases in the horizontal direction.

1. Reflections
2. Conclusion
   1. Limit the methods in the pixel-level operations, especially without recognizing any geometric shapes.
   2. Not qualitatively different from the previous work of my advisor. I just add more transformations and analogies to see how far we can go in this way.
   3. Implications are twofold:
      1. For AI, Considering the nature of this kind of intelligence tests like Raven’s, it is an unavoidable issue …
      2. For human, this model serves as a demonstration of how analogical ability contribute to indifference in the Raven’s test.
   4. Future work