Project 3

Xuejian Liu

Introduction

The approach the agent uses in Project 3 to solve Ravens Progressive Matrices (RPM) 3x3 basic problems D and E is an extension of the designed method described in Project 2 to answer the basic problems C. Due to the success of pure visual reasoning in identifying the answers to 11/12 of the basic problems C, more patterns were coded in the previous agent in order to solve the problem sets D and E. Multiple problem solving methods including case based reasoning, classification, logic and analogical reasoning are applied to attempt to determine the correct answers.

Agent design

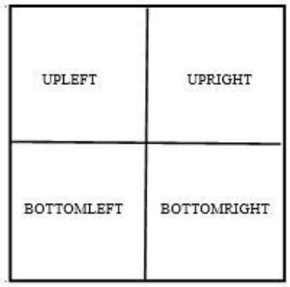
The success in identification of the correct answers of the majority of basic problems D and E gives me better understanding the whole process of RPM problems solving. Basically, the agent was designed sequentially to solve the RPM 2x2 and 3x3 problems and separate solvers were constructed for 2x2 and 3x3 sets. The overall solutions are a combination of verbal and visual representations. In project 3, we did not spend time on improving the performance of the agent on 2x2 sets due to limited time. The AI agent mainly relied on verbal reasoning to answer 2x2 problems, which was described in Project 1 with details. Verbal agent which is object based offers some advantages over purely visual reasoning 3x3 agent that is generally figure based though easier implementation of visual representation was found and we will discuss later on this point.

Solving 3x3 RPMs

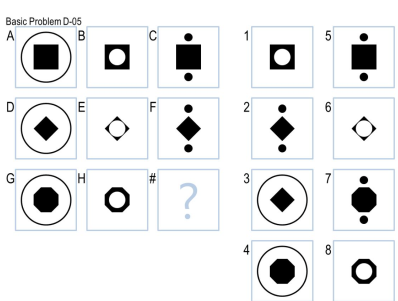
Since the focus of Project 3 is 3x3 problems, we would like to give a clear picture of the solution to these kinds of problems. (The design part in Project 2 is rewritten for Project 3 reflection due to feedback from TA concerning not very clear description.)

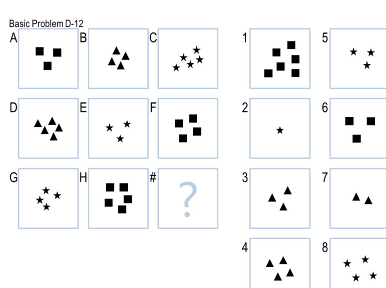
The basic idea is, for a given 3x3 problem, there are fixed numbers (8) of figures (A, B, C, D, E, F, G, H) in the problem. The changes in rows and columns may fulfill some rules (patterns). A couple of different patterns are predefined and the 8 figures in a problem are analyzed either horizontally or vertically to try to match it with one of the pattern and the details are shown below.

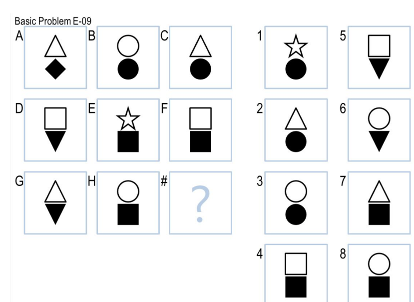
For each 3x3 problem, the figures 1-8 from the provided files are loaded into BufferedImage (Java). This loading does not focus on the extraction of the actual image attributes (shape, size, rotation, etc.), but the calculation of the total number of black pixels in images (numA, numB, numC, numD, numE, numF, numG, numH). Further, to better identify the changes in rows and columns, a image is generally separated evenly into 4 regions (‘UPLEFT’, ‘UPRIGHT’, ‘BOTTOMLEFT’ and ‘BOTTOMRIGHT’) and the corresponding number of pixels in each of the four regions are calculated. The number of pixels for: the left part of the figure (upleft+bottomleft), the right part (upright+bottomright), the up part (upleft+upright) and the bottom part (bottomleft+bottomright) are also readily obtained.



After reviewing the problem sets C, D and E, different kinds of transformation logic exist across the rows and columns. There are simple cases of logic, for example, the total number of pixels are the same in each figure in one row (basic problem C-01) or the total number of pixels in one row or columns consistently increase (basic problem C-03). A predefined pattern with similar total number of pixels in every image in each row separately is appropriate for solving problem C-01. In this pattern, the agent calculated all the number of pixels in the candidate figures to find '3' to be the right answer (num3 = numG or numH). An error (0.05) is allowed when comparing the similarity. For some problems, more complicated transformation exists. For example, in basic problem D-05, the difference of the number of black pixels in figures in one column is fixed and a pattern with entry criteria numA - numD = numB - numE = numC - numF was selected. The difference of total number of black pixels in the correct answer and numF should be close to numH - numE to fit the pattern and '7' is given to be the right answer. Basic problem D-12 is a special case and a particular pattern was programmed in order to solve the problem. The pattern has features like numA / 3 = numF / 4 and numB / 4 = numD / 5 was initially constructed and the agent chose this pattern for D-12 and looped over all candidate figures using num / 3 = numD / 5 as the rule to find '3' be the answer. A pattern with entry rules numUpA = numUpC = numUpG, numUpB = numUpH, numUpD = numUpF, numBottomD = numBottomG, numBottomE = numBottomH, numBottomB = numBottomC and numBottomE = numBottomF was identified for basic problem E-09 and the candidate figure '7' with numUp and numBottom close to numUpC and numBottomG respectively is the answer.







The whole problem solving is a transformation pattern match process. The pixel differences from A→B→C to D→E→F and from A →D →G to B →E →H in each region are readily accessed and then the appropriate pattern is selected to solve the problem. In certain cases, both horizontal and vertical directions can be matched so that either one can be chosen, while in some others, only in one direction, a pattern can be found. In such case, the matched direction is selected. Subsequently, the corresponding 3rd column or row is matched by searching all candidates and the closest one which can form the expected pattern is identified to be the appropriate answer.

Pattern ordering

It is worth mentioned that the patterns are prioritized, for instance, a reflection pattern (basic problem C-07) and an 'all-same' (basic problem C-01) pattern . Obviously a shape and itself copied to another, can also be in perfect reflection, however we analyze the provided problems and find that in most cases, if a problem follows the 'all-same' pattern, it can also be solved using the reflection pattern, but not vice versa. As such, reflection pattern is determined to have a higher priority than 'all-same' pattern.

More and more problem types of problems are required to solve and it is difficult to find general solutions, which is also a weakness when using purely figure based agent. More patterns were constructed in Project 3 based on the Project 2 in order to solve the problems of the basic D and E sets though the agent we designed in project 2 is capable of answering 8/12 of the basic D sets. This raised another challenge: how to order the patterns reasonably? Due to the previously great efforts to prioritize the patterns in Project 2, we established a solver for E problem set separately to avoid the complexity. Much time is required to address this issue.

Results

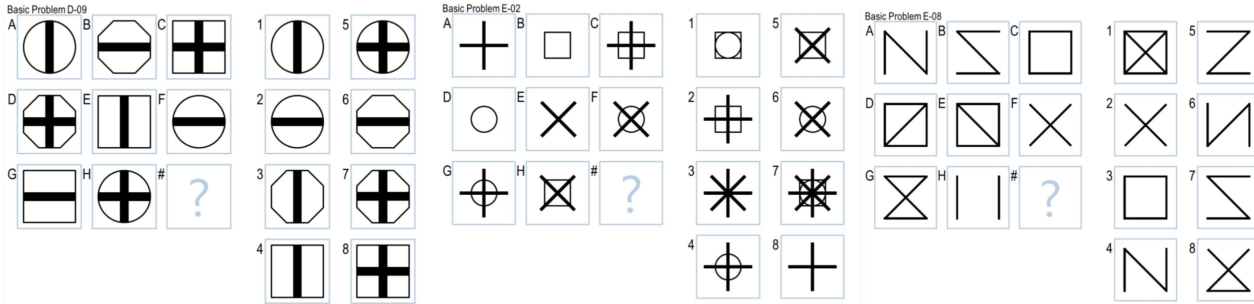
The overall performance of the agent on basic and challenge problems are tested and the results of the test are as shown below:
Basic Problems B: 10/12
Challenge Problems B: 2/12
Basic Problems C: 11/12
Challenge Problems C: 4/12
Basic Problems D: 11/12
Challenge Problems D: 4/12
Basic Problems E: 10/12
Challenge Problems E: 2/12

Efficiency

The overall run time for solving all problem sets of B, C, D and E is within seconds. The efficiency of the agent is directly proportional to the complexity of the input problem. Problem with more transformation features tend to take more time as there are more calculations involved.

Failure and Generality

For the 3x3 solver, the mistakes the agent made are because it is challenging to generalize a meta to define the patterns. In the case of basic D-09, the total number of black pixels in rows or in columns should be similar. It turned out that my agent could not differentiate answer '1' and '3' and it is assumed that there are pixel differences in rows or in columns in the original images of the problems. By tuning the threshold of similarity to larger number, the problem can be solved. However, this causes the failure of other problems. For basic E-02, I found it difficult to define an aggregation pattern in row and column to solve the problem due to the bigger extent of overlap in the third row and column. While for basic E-08, the pattern of merging and removal is difficult to express with algorithm.



Only a little time was spent on challenge problem sets, it is envisioned that more patterns or the tweak of the existing patterns are required to solve the problems.

Though it is feasible to design an agent to solve most of the basic problems based on the comparison of the number of black pixels, the figure based visual method we established is potentially lack of generality to solve the problems with object based transformations. The reasoning logic in Project 2 was classified based on the property of the problem in basic C problem set. It was found that when new problems with different transformation property such as the E problem set was given, the agent may skip or fail some of the problems. With the increasing number of types of problems, we have to code more patterns in the agent to answer them. In some cases, a number was used in the agent to calculate the shape or fill changes related pixel differences, for example, 1.1 was determined to be the scale up of a circle or a square in basic D-08. The some-what arbitrary number makes it difficult to determine the general enlarged shape size in other problems. With more problem examples given, the uncertain number can be tuned to be more reasonable for general problems. The hard-coded patterns in the agent was thought to be the reason of poorer performance in the test problems in comparison to basic problem sets.

Improvement

I would like to mention that I would try to solve the 2x2 problems using visual reasoning to make a comparison of the different methods given unlimited time. Though it seems that visual presentation is more appropriate to answer 3x3 sets than 2x2 because more transformation information are provided and it is easier and more accurate to define a pattern, the visual reasoning is generally easier to implement than the verbal counterpart.

There are scopes for improvement in 3x3 solver: the more meta characteristic we define in a pattern, more precise the agent recognition would be, especially for those similar patterns. To separate the image into 9 regions or even 16 regions and track the changes would be a good idea. More patterns can also be composed for better performance. Reasonable priorities to each pattern we predefine can be determined and we spent a lot of time on tweaking it for enhanced agent performance.

In terms of generality, we envisage great efforts in the construction of a library of different transformation patterns in the agent's knowledge base with more general algorithm to identify the changes. More problems sets are in need to construct overall rules for similar patterns. Once the generality is enhanced, more problem types are handled. With insufficient knowledge in Java, it is also assumed that if the the 3x3 solver can identify the changes based on the object level, it should be easier to find general solutions than the current figure based methods.

Relation to human cognition

Human cognition can understand the problem, then apply or expand his/her previous knowledge and experience to solve the problem while ‘knowledge’ and ‘rules’ are encoded to the agent in the design in order to find an answer. Utilizing the visual reasoning, the agent mimic the process human follows: reading images, partitioning the image into different regions, calculating the changes and identifying the appropriate patterns of transformations.

Humans also do a good job at solving similar problems using knowledge gained previously. In a similar way, the agent can determine the answer to similar problems by applying the programmed knowledge and logic included in the old cases. The agent designed in Project 2 was found to be capable of solving some problems in the basic D and E. Humans can also learn from the mistakes and refine the knowledge to successfully answer similar problems. Likewise, with the modification of the code, repaired knowledge base allows the agent to improve the performance in similar problems.

However, the agent is limited to its programmed knowledge/rules. For the visual reasoning based 3x3 solver, only if the pattern related algorithm is defined, the problem can be solved. In the cases of basic E-02 and E-08, it turned out to be difficult to determine the pixel differences to define the pattern and the agent hence failed in the problems. In such cases, human can instantly answer the problems without thinking of calculation of the pixel changes related to the problem but by identifying the shape changes. Humans are also excellent in using combinations of methods, for example, to determine the answer of some problems using the shape and pixel transformation simultaneously.

In some cases, the agent can be designed to present advantages over humans. The agent can recognize the differences between images with only slight differences, which may not be easily found by human vision. This can be both a strength and also a weakness. Just take the basic D-09 for example, it is apparent to be an aggregation pattern and human can readily decide ‘3’ is the correct answer. Theoretically, it should be easy for the agent to find only ‘3’ meets the criteria of the pattern. However, due to the pixels differences in the original figures were calculated by the agent to be not identical in rows and columns, it eventually turned out that the agent can not make an appropriate choice. Another virtue of the agent is, by coding with improved rules, the agent can process the RPM problems more efficiently than human.