Computers in Human Behavior 41 (2014) 319–326

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

How people do relational reasoning? Role of problem complexity and domain familiarity

Shikhar Kumar [a,c,*], Iliano Cervesato [b], Cleotilde Gonzalez [c]

[a] School of Information: Science, Technology and Arts, The University of Arizona, Tucson, AZ, United States
[b] School of Computer Science, Carnegie Mellon University in Qatar, Doha, Qatar
[c] Department of Social and Decision Sciences, Carnegie Mellon University, Pittsburgh, PA, United States

ARTICLE INFO

Article history:
Available online 1 November 2014

Keywords:
Relational reasoning
Problem solving
Spreadsheet
Domain familiarity
Problem complexity

ABSTRACT

The goal of this paper is to study how people do relational reasoning, such as selecting the grade of all students in a class with GPA (Grade Point Average) greater than 3.5. Literature in the field of psychology of human reasoning offer little insight as to how people solve relational problems. We present two studies that look at human performance in relational problems that use basic relational operators. Our results present the first evidence toward the role of problem complexity on performance as determined by the accuracy and discrimination rates. We also look at the role of familiarity with tabular representation of information, as found in spreadsheets for example, and other factors for relational reasoning, and show that familiarity does not play a significant role in determining performance in relational problem solving, which we found counterintuitive.

1. Introduction

Nowadays, data are more easily accessible than ever, yet support for deriving interesting consequences from base data is often unavailable, too expensive, or too technical for many users. For example, a student may have access to prerequisite listings and expected offering dates of courses but have no way to sieve through possible course sequences unless the college provides a dedicated tool. Similarly, an investor may know the instruments held in his mutual fund portfolio but have no easy way to unravel them and reveal his exposure to a specific industry or company. In all cases, manually inferring useful information from raw data is time consuming and error prone, a situation that often results in bad decisions, suboptimal plans, or missed opportunities. In fact, there is currently no simple and general application that empowers users to compute useful inferences on raw data.

Cervesato (2007, 2013) addressed this problem by drawing inspiration from a type of automated data inference that is immensely popular: the spreadsheet. Applications such as Microsoft Excel and others are readily available and allow users to routinely perform complex custom calculations on numerical data. The spreadsheet's clever interface makes it easy to use productively with little or no training. However, none of the above data manipulation problems is expressible in today's spreadsheets. The approach investigated in the cited work, which was dubbed NEXCEL in (Cervesato, 2007), remedies this situation by extending the spreadsheet paradigm to enable users to define useful forms of inference among their data. It allows the student, for example, to download data about course prerequisites and offerings into his favorite spreadsheet, and write a "formula" that calculates all possible course sequences. The investor can similarly see the individual stocks in his portfolio and determine his actual exposure.

These "formulas" combine not numbers but *relations* (for example the relation that associates courses to each of their prerequisites, or the relation between mutual funds and publicly traded companies). Just like traditional spreadsheets leverage the ability of their users to capture numerical inferences using numerical formulas, NEXCEL asks users to express *relational reasoning* using these relational formulas. However, little is known about how people do relational reasoning. Clearly some people are very good at it (e.g., database programmers). But how natural is it for the rest of us? Which relational constructs do humans find easy to use? Which ones lead us to make more mistakes? How should a language of relational formulas be constructed to capture most immediately the way we do relational reasoning?

We did not find answers to these questions in the literature. The closest studies we are aware of are Johnson-Laird's analysis of how people fare with various forms of logical inferences (Johnson-Laird,

* Corresponding author at: School of Information: Science, Technology and Arts, The University of Arizona, PO Box 210077, Tucson, AZ 85721, United States.

E-mail addresses: shikhark@email.arizona.edu (S. Kumar), coty@cmu.edu (C. Gonzalez).

http://dx.doi.org/10.1016/j.chb.2014.09.015
0747-5632/

1983, 2006; Johnson-Laird & Byrne, 1991; Johnson-Laird, Byrne, & Schaeken, 1992) and Oaksford and Chater's probabilistic approach to human reasoning (Oaksford & Chater, 2001, 2007, 2009).

The analysis of human performance in reasoning tasks have shown that people make large and systematic errors, which are not random (Evans et al., 1993; Manktelow, 1999), suggesting that humans might be irrational (Stein, 1996; Stich, 1985). These observations have led to numerous studies, with several formal models of human reasoning. Some of the most well known approaches involve the comparison of human performance against formal logic.

In the logical analysis of human reasoning, two major approaches have been utilized: mental-logic approach (Rips, 1994) and mental-model approach (Johnson-Laird & Byrne, 1991). Both these approaches argue that the systematic deviations from logic in deductive inference tasks represent unavoidable performance errors, which stem from limited working memory. Given humans have limited working memory and other cognitive abilities, it restricts their reasoning abilities. Thus, in principle humans are rational but in practice they are constrained by cognitive limitations.

In contrast to logic based approaches, a more recent approach to the analysis of human reasoning is the probabilistic approach (Oaksford & Chater, 2007). It posits that everyday reasoning is probabilistic and the reason why people make errors in logical tasks conducted in the laboratories is because they generalize these everyday strategies to the laboratory. Oaksford and Chater argue that logic is inadequate to account for everyday reasoning and probabilistic approach is more promising (Oaksford & Chater, 2001). This approach has been applied to several core areas of the psychology of reasoning: conditional inference (Oaksford, Chater, & Larkin, 2000; Schroyens, Schaeken, Fias, & d'Ydewalle, 2000; Schroyens, Verschueren, Schaeken, & d'Ydewalle, 2000), syllogistic reasoning (Chan & Chua, 1994; George, 1997, 1999; Liu, 1996; Stevenson & Over, 1995), and Wason's selection task (Oaksford & Chater, 1994, 1998). In all these areas it has been shown that probabilistic approach offers a better explanation to human performance than more traditional normative approaches.

Here we will not argue which approach is better; rather we will consider what both the logical and probabilistic approaches to human reasoning have to offer with regards to our understanding of relational inference. Both offer two different explanations of the performance of humans in inference tasks: (1) Limitation of cognitive abilities (2) People employ strategies from everyday reasoning in the laboratory. We posit that limitations of cognitive abilities will play a significant role in relational inference. However, we are not quite sure if the strategies from everyday reasoning will be utilized in relational inference and how they might affect the performance.

Answering these questions is critical to the development of tools like NEXCEL, and more generally to understand the ways in which humans carry out relational inferences, how relational inferences compare to numerical inferences; and how they would perform both types of inferences with spreadsheet capabilities. To this end, we have designed a series of studies whose purpose is to answer precisely these questions. In the present paper, we report on two studies that explore our subjects' ability to carry out the most basic forms of relational inference. In the first experiment, we study how humans perform four elementary relational operations: projection, union, difference and join (which include more general forms of selection). We did so using a traditional spreadsheet as a visual proxy. The second experiment aims at investigating how more complex relational operations are resolved. We combined various simple operations to test human performance. Future experiments will explore more complex operators (e.g., recursion) and combinations (e.g., nested negations), and gauge the subjects' ability to express the relational reasoning patterns needed to solve a problem in a variety of relational languages.

A *relation* can be visualized as a table consisting of rows and columns. Each column, or *attribute*, holds data with a consistent meaning (e.g., the grade of a student, or the name of a mutual fund). Each row, or *record*, contains specific data in the relation, for example the name, grade, and major of a specific student in a class. Relational inference computes new relations on the basis of relations we already know, for example the students with a GPA (Grade Point Average) greater than 3.5 together with their major. Relations do not contain duplicate records. Any relational inference can be obtained by combining a small number of elementary relational operations (in the same way as any arithmetic expression is based on addition, subtraction, etc). In our experiments, we relied on four of these elementary operations: projection, union, join and difference.[1] We will now give details of these operations.

- *Projection* simply deletes some attributes from a relation (and removes any duplicate record that may ensue). For example, a professor may need to make a list of student names and their respective grades for some exam. However, he only has a full grade sheet of the students, with their majors, and other information. Removing the unwanted columns is a use of projection.
- *Union* combines two (or more) relations with the exact same attributes into a single relation. For example, the professor may have two grade sheets, one for each section of the same class, and may need to look at the grades of all the students in the class. This task of combining both grade sheets into one is a form of union.
- *Join* is more complex: Given two sets of records with a common attribute, join combines the records that share the same value for this attribute. For example, if a professor has a list of students and the classes they take and another list of students and the sports they play, she may need the list of all students with their respective classes and sports. Here the task can be accomplished by joining the two sets of records based on student names.
- *Difference* retains the records that are in one relation but not in a second one. Like union, both relations should have the exact same attributes. For example, a professor with separate grade sheets for the two sections of her class may want to examine the performance of the students coming to the morning section only (knowing that some students attend both the morning and the afternoon section). The operation she would use to do so is difference: she wants the record of the students in the morning section that do not occur in the afternoon section.

We expected that humans would find these relational operations easy to accomplish, and that some of the more complex operations such as join would result in more mistakes and lower accuracy. We also expected that the participant's familiarity with tabular representation of information (e.g., in spreadsheets) and other relevant topics like databases, programming, logic, mathematics, etc, would help in solving problems with more complex operations.

[1] Traditionally, join is itself decomposed into selection and Cartesian product. The latter is rarely used in isolation, and therefore would have led to artificial experiment tasks. Selection, which is commonly used in practice, becomes a special case of join. Modern presentations include recursion as an additional relational operation. We believe it is significantly more complex than the other operations, and therefore decided to dedicate a separate study to it. Every relational inference can be expressed as a combination of union, projection, selection, Cartesian product, difference and recursion (Cervesato, 2013).