Can Binary Decision Diagrams predict concept complexity in human learning?

Bachelor's Project Thesis

Bekli Shtrepi, s3170683, b.shtrepi@student.rug.nl
Supervisors: Dr B. R. M. Gattinger

Abstract: This project is about predicting the difficulty of concept learning in humans using Binary Decision Diagrams. The subjective difficulty of learning has been studied in the past using multiple other methods, including propositional logical formulas (Feldman, 2000). In this paper, we take a look at previous research data, and we use those results to see whether BDDs are able to make a correct prediction. We find a moderate correlation of 67% between the depth of the BDD and the Boolean complexity of the corresponding concepts. We also find a much stronger correlation of 86% between the number of nodes of the BDD and the Boolean complexity of the concept.

1 Introduction

1.1 Predicting Concept Learning Complexity

In the field of human concept learning, one of the prominent and unsolved problems has been determining the factors that dictate the difficulty of learning a concept. That is to say why are some concepts seemingly easy to learn, while others appear harder, more complex, or incoherent.

Predicting the difficulty of concept learning in humans has been a topic that did not receive much attention until the 1960s. During this time period, there was a large number of studies conducted on this topic (Bourne, 1970). However, none of them were able to answer this question. Furthermore the move from seeing concepts as logical rules to seeing them as prototypes (Posner and Keele, 1968; Rosch, 1973) has left this question still unanswered (Osherson and Smith, 1981; Medin and Smith, 1984; Komatsu, 1992).

One particular study by Shepard, Hovland, and Jenkins (1961) investigated concepts of three features with four positive and four negative examples. These concepts fall into six different types labeled I through VI, denominated as the SHJ types. All concepts with three features and four positive examples are isomorphic to one of the six types. At the same time, none of the concepts of one type are isomorphic to another type. This makes the classification a complete one. In their study, Shepard et al. (1961) discovered that the six types differed in difficulty in the following order: I<II<[III, IV, V]<VI, where III, IV, V have approximately the same level of difficulty. Type I was the easiest to learn or understand and VI the most difficult. The findings of this study have been replicated since (Nosofsky, Palmeri, and McKinley, 1994), however, the question of why some concepts were easier to learn and understand than others still lacked an elegant or simple answer.

The question was finally answered by Feldman (2000) where the author exhaustively examined concepts and their complexity. The study discovered a link between the difficulty of learning certain concepts and the complexity of the logical formulas used to represent the concepts.

In this study, we will be looking at the BDD representation of those same concepts from Feldman (2000), and their characteristics. We will then conduct a number of tests in order to determine whether there exists a link between the properties of BDDs and the human learning complexity of the concept. We will be looking at the correlation between two different properties of BDDs (depth and number of nodes) and the Boolean complexity of the concept.

In order to check the correlation, we will be using Pearson's product-moment correlation test. This test measures the linear relationship between two linear variables and it returns, as a result, a number between -1 and 1 (Chee, 2013). When the result is 1 that means there is a complete positive correlation between the variables. When the result is -1 there is a complete negative correlation between the variables and in the case that it is 0 that means that there is no correlation between the two. The assumptions of the test are that the correlation is linear and that it is normally distributed. These assumptions are tested in section 3.1.

1.2 Binary Decision Trees

In order to understand binary decision diagrams we first need to understand binary decision trees and the boolean language.

The boolean language refers to a subset of algebra used to create true/false statements. The boolean expressions we will be focusing on in this paper will be composed of boolean-typed values (which are either true or false) and the boolean operators *and, or* and *not* ($\wedge, \vee, \neg$ respectively). In BNF we would have:
<atom> ::= <identifier> | '(' <formula> ')'
<literal> ::= <atom> | ¬ <atom>
<formula> ::= <literal> {∧ <literal> | ∨ <literal>}

These expressions correspond to propositional formulas in logic. Now we will look at boolean decision trees.

In order to represent the formula

$$a \vee b \tag{1.1}$$

we would have as the root node a with two lines coming from it leading to two other nodes `b`. One of the lines will be dashed (for False) and the other will be solid (for True). Then we would have another two lines coming from each of those nodes to represent the two values the variables can have. Finally, the leaf nodes will be either ones or zeros. We can see a visual representation in Figure 1.1

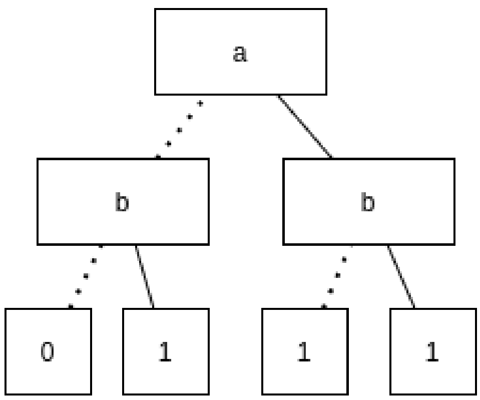

Figure 1.1: Binary Tree of formula 1.1

A BDT is a binary tree such that each node represents one of the variables in a boolean expression, and each line connecting two nodes is one of the two possible values the variable above can take.

This data structure has several benefits, such as canonicity. This means that if we test the variables in a fixed order, then the resulting BDT will be unique. However, the biggest problem is the size of the structure. As we can see in Figure 1.1 with just two variables there are a total of 3 intermediate nodes and 4 leaf nodes. This representation will have $2^n - 1$ intermediate nodes and 2^n leaf nodes, where n is the number of variables and the leaf nodes are ones and zeros.

1.3 Binary Decision Diagrams

Binary Decision Diagrams (BDDs) are a type of data structure that are used to represent boolean functions as graphs (Bryant, 1992). Boolean functions take booleans as inputs and produce a boolean as output. BDDs are similar to Binary decision trees, however, they remove redundant operations.

$$a \wedge b \tag{1.2}$$

For example, in the case of formula 1.2 when a is 0, then regardless of b, the result will be 0. The resulting BDD will be as seen in Figure 1.2. In this case, testing variable b on the left branch would be redundant since it would have no effect on the outcome.

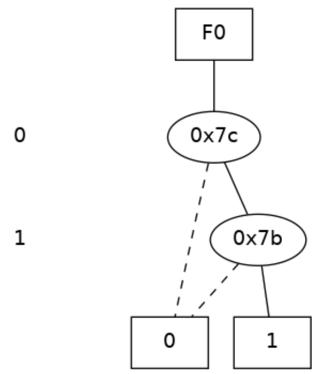

Figure 1.2: Bdd for formula 1.2

First I will explain the notation seen in figure 1.2. The starting variable at the top (F0) is not an actual node of the BDD. It simply denotes the start of the diagram. The names inside of the nodes represent the variables in the original propositional formula. In our case, for Figure 1.2, "0x7c" represents variable "a" while node "0x7b" represents variable "b". The naming scheme was the default one in the CUDD package (Somenzi, 2015) used to create the BDDs. The numbers seen on the left of the nodes denote the depth of the nodes.

BDDs also offer the advantage of sharing identical subtrees. This can be seen with Formula 1.3 and its respective diagram, Figure 2.2. In this case, we see some of the nodes share the same subtrees. This allows us to represent the same result but with a much smaller size since we are not repeating the same operations many times over.

If we look at the BDD of the formula from 1.1, the diagram would be as we see in Figure 1.3 As we can see this representation is smaller in size than the binary tree.

Although in the case of formula 1.1 the difference is not massive between the two different representations, it shows us the differences between the two structures. In cases with longer formulas, the

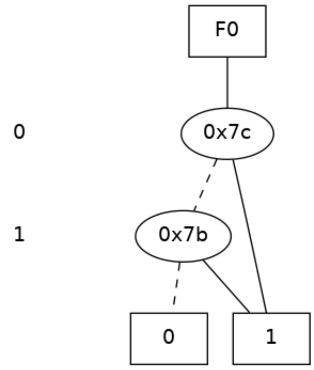

Figure 1.3: BDD of formula 1.1

difference can be much more significant. We can see an example of this in Figure 2.2 for formula 1.3. We see the formula here is much longer and the diagram is larger as well.

BDDs can be considered as a compressed representation of sets or relations. However, unlike other compressed representations, operations are directly performed on the compressed state of the diagram, as opposed to first decompressing it.

In the worse case for a BDD, the number of nodes will still be exponentially bigger than the number of variables. However, in most cases, BDDs will be much smaller than binary decision trees.

1.4 Previous Research

As previously stated, the concepts that will be tested are the same ones found in the Feldman (2000) paper. The concepts were translated into logical formulas. The formulas and their respective complexity can be seen in Table 2.1. Note that the formulas in the table are in the reduced state, whereas formula 1.3 is not. The reduced form for formula 1.3 is in row 40 of the table.

$$\begin{aligned}(\neg a \wedge \neg b \wedge \neg c \wedge \neg d) \vee (\neg a \wedge \neg b \wedge c \wedge d)\\ \vee(a \wedge b \wedge \neg c \wedge \neg d) \vee (a \wedge b \wedge c \wedge d)\end{aligned} \tag{1.3}$$

The reduced formulas are logically equivalent to the non-reduced formulas. The reduction is used here in order to have a more compact and more readable format. As can be seen in formula 1.3, these formulas can get quite long with redundant operations. Since the two different versions are logically equivalent their BDDs will also be identical.

2 Method

2.1 Program

In order to complete the research, we will need a method to transform the propositional formulas into BDDs. This is done by using a simple program that takes as input a ”formulas.txt” file and then returns the appropriate BDD into a ”.dot” file.

The ”formula.txt” file contains the 41 formulas that will be tested in this paper. The program can accept more formulas as long as they are in DNF form. The ”.dot” files are in the DOT language. This language is used to describe graphs.

These can be visualized by having the *graphviz* library installed onto your computer (Ellson, Gansner, Koutsofios, North, and Woodhull, 2004). This program is based on the *Colorado University decision diagram (CUDD)* package (Somenzi, 2015)

The code can be downloaded from GitHub: `https://github.com/brewmaster011/bdd_generator`

2.2 The D[P] hierarchy

In his paper Feldman (2000), described a method to encapsulate different formulas into different families. He made the distinction between families by separating the concepts by P positive examples over D binary features.

Any concept that falls within this description is logically equivalent to the disjunction of P objects, each of which is a conjunction of D features. This form is what is known as Disjunction Normal Form (DNF) (Feldman, 2000).

$$(\neg a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge \neg b \wedge c) \tag{2.1}$$

For example, we can then see that formula 2.1 belongs to family 3[2]. We see that there are two disjuncts of constituent objects, each a conjunction of three features. If we look at formula 1.3 we can see that this formula belongs to the 4[4] family. Here we have the disjunction of four constituent objects, each of which a conjunction of four features.

By changing the values of D and P we get different families. The family tested in the paper by Shepard et al. (1961) was the 3[4] family. While the families tested by Feldman (2000), that will also be tested in this paper, are: 3[2], 3[3], 3[4], 4[2], 4[3] and 4[4].

This hierarchy extends infinitely on values of D, while the values of P can be less, or equal to 2^{D-1}

2.3 Formulas

Among the six different families of formulas, one of them is from the study of Shepard et al. (1961). The other five were added by Feldman in order to extend the range of propositional concepts that were being tested. The list of concepts that are tested here is an exhaustive list.

The six families make up 41 distinct formulas, with the number of variables being between two and four. The formulas are composed of negations, conjunctions, and disjunctions. We can see the reduced formulas in table 2.1.

2.4 Transforming the formulas into BDDs

In order to transform the formulas into Binary Decision Diagrams we will be using a BDD package for the C language called CUDD (Kebo, 2017). The formulas are fed into the program using a text file. The program then creates a file with the graph for the appropriate formula and it also prints some helpful information about each of the diagrams.

For example, for the following formula: $(\neg a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge \neg b \wedge c)$, we would have figure 2.1

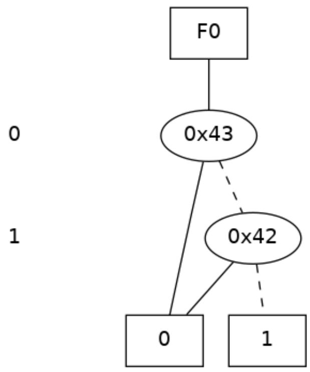

Figure 2.1: BDD of Formula 2.1

This is one of the shortest formulas that we will be looking at. For contrast we can look at one of the more complicated formulas (1.3) and it's corresponding BDD (figure 2.2).

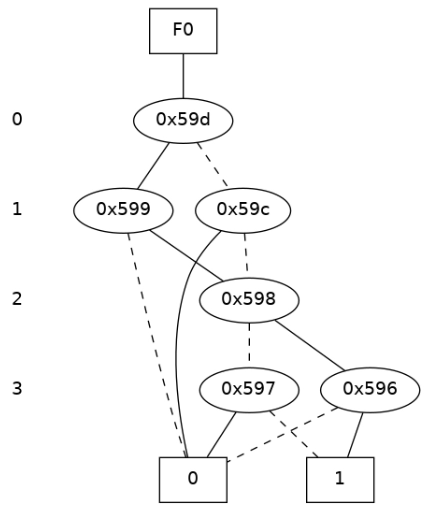

Figure 2.2: BDD of Formula 1.3

As we can see for this one, the formula is longer (also more complex) and the BDD representation is also longer compared to the previous example.

2.5 Dependent Variables

The two main variables we observe about the BDDs are the depth and the number of nodes. We can see that depth on the left side of the BDD representation starting from zero. We see that in the simpler formula 1.1 the maximum depth is one. While in the more complicated formula 1.3 the maximum depth is three.

If we look at the number of variables we see that with the simpler formula we have fewer nodes for the simpler formulas. For example, for formula 1.1 we only have two nodes. Whereas for formula 1.3 we have six nodes.

In order to see whether there is a correlation between these values and the complexity of a concept, I will conduct a correlation test between these variables and the difficulty level discovered in the Feldman (2000) paper. Each of the concepts in that paper is accompanied by a Boolean complexity which can also be seen in the formula table 2.1.

Two separate tests will be carried out, one for the depth of the BDDs and one for the number of nodes in the BDDs. Simply looking at the diagrams we see more of a correlation between the number of nodes and the difficulty of the concepts. The BDD depth, although greater with more complex concepts, does not change much overall.

3 Results

3.1 Preliminary testing

Before the correlation tests, we need to test some assumptions. These assumptions need to be fulfilled in order for us to get accurate and valid results. We need to check whether the covariation is linear and whether the data is normally distributed. The covariation can be seen in Figures 3.8 and 3.4 below. To check whether the data is normally distributed the Shapiro-Wilk normality test will be used.

Firstly we check if the covariation is linear. We can see in both Figures 3.8 and 3.4 that the lines are more or less straight with minimal curvatures. We can then say that this assumption is satisfied.

The Shapiro-Wilk normality test tells us whether the data is significantly differently distributed as compared to a normal distribution. After conducting the test we see that all of our data is normally distributed and this assumption is fulfilled as well.

Given that the data set is quite small, in order to make sure the data is normally distributed we will also take subjectively at the mode, median, and mean. In a normal distribution, these three measures should be around the same value, and this is what we see with our variables. Therefore we can say confidently that the data is normally

Table 2.1: Table of Minimal Formulas

Nr.	Minimal Formula	Complexity	BDD Nodes	BDD Depth
	Family 3[2]			
1	$\neg a \wedge \neg b$	2	2	1
2	$\neg a \wedge ((\neg b \wedge \neg c) \vee (b \wedge c))$	5	4	2
3	$(\neg a \wedge \neg b \wedge \neg c) \vee (a \wedge b \wedge c)$	6	5	2
	Family 3[3]			
4	$\neg a \wedge \neg(b \wedge c)$	3	3	2
5	$(\neg a \wedge \neg b) \vee (a \wedge b \wedge \neg c)$	5	4	2
6	$(\neg a \wedge ((\neg b \wedge \neg c) \vee (b \wedge c)) \vee (a \wedge b \wedge \neg c))$	8	5	2
	Family 3[4]			
7	$\neg a$	1	1	0
8	$(a \wedge b) \vee (\neg a \wedge \neg b)$	4	4	2
9	$(\neg a \wedge \neg(b \wedge c)) \vee (a \wedge \neg b \wedge c)$	6	5	2
10	$(\neg a \wedge \neg(b \wedge c)) \vee (a \wedge \neg b \wedge \neg c)$	6	4	2
11	$(\neg a \wedge \neg(b \wedge c)) \vee (a \wedge b \wedge c)$	6	5	2
12	$(a \wedge ((\neg b \wedge c) \vee (b \wedge \neg c))) \vee (\neg a \wedge ((\neg b \wedge \neg c) \vee (b \wedge c)))$	10	5	2
	Family 4[2]			
13	$\neg a \wedge \neg b \wedge \neg c$	3	3	2
14	$\neg a \wedge \neg b \wedge ((\neg c \wedge \neg d) \vee (c \wedge d))$	6	5	3
15	$\neg a \wedge ((\neg b \wedge \neg c \wedge \neg d) \vee (b \wedge c \wedge d))$	7	6	3
16	$(\neg a \wedge \neg b \wedge \neg c \wedge \neg d) \vee (a \wedge b \wedge c \wedge d)$	8	7	3
	Family 4[3]			
17	$\neg a \wedge \neg b \wedge \neg(c \wedge d)$	4	4	3
18	$\neg a \wedge ((\neg b \wedge \neg c) \vee (b \wedge c \wedge \neg d))$	6	5	3
19	$(\neg a \wedge \neg b \wedge \neg c) \vee (a \wedge b \wedge c \wedge \neg d)$	7	6	3
20	$\neg a \wedge (\neg b \wedge ((\neg c \wedge \neg d) \vee (c \wedge d) \vee (b \wedge \neg c \wedge d)))$	9	6	3
21	$(\neg a \wedge \neg b \wedge ((\neg c \wedge \neg d) \vee (c \wedge d))) \vee (a \wedge b \wedge \neg c \wedge \neg d)$	10	7	3
22	$(\neg a \wedge \neg b \wedge ((\neg c \wedge \neg d) \vee (c \wedge d))) \vee (a \wedge b \wedge \neg c \wedge d)$	10	7	3
	Family 4[4]			
23	$\neg a \wedge \neg b$	2	2	1
24	$\neg a \wedge ((\neg b \wedge \neg(c \wedge d) \vee (b \wedge \neg c \wedge \neg d))$	7	6	3
25	$\neg a \wedge ((\neg b \wedge \neg(c \wedge d) \vee (b \wedge \neg c \wedge d))$	7	5	3
26	$\neg a \wedge ((\neg b \wedge \neg(c \wedge d) \vee (b \wedge c \wedge d))$	7	6	3
27	$(\neg a \wedge \neg b \wedge \neg(c \wedge d)) \vee (a \wedge b \wedge \neg c \wedge \neg d)$	8	6	3
28	$(\neg a \wedge \neg b \wedge \neg(c \wedge d)) \vee (a \wedge b \wedge \neg c \wedge d)$	8	6	3
29	$(\neg a \wedge \neg b \wedge \neg(c \wedge d)) \vee (a \wedge b \wedge c \wedge d)$	8	7	3
30	$\neg a \wedge ((b \wedge c) \vee (\neg b \wedge \neg c))$	5	6	3
31	$(c \wedge \neg d \wedge ((\neg a \wedge b) \vee (a \wedge \neg b))) \vee (\neg a \wedge \neg b \wedge \neg c)$	9	6	3
32	$(\neg a \wedge ((\neg b \wedge \neg c) \vee (b \wedge c \wedge \neg d))) \vee (a \wedge \neg b \wedge c \wedge d)$	10	8	3
33	$(b \wedge c \wedge \neg d) \vee (\neg a \wedge \neg b \wedge \neg c)$	6	6	3
34	$(b \wedge c \wedge ((\neg a \wedge \neg d) \vee (a \wedge d))) \vee (\neg a \wedge \neg b \wedge \neg c)$	9	8	3
35	$(a \wedge b \wedge c) \vee (\neg a \wedge \neg b \wedge \neg c)$	6	5	2
36	$\neg a \wedge ((b \wedge ((\neg c \wedge d) \vee (c \wedge \neg d))) \vee (\neg b \wedge ((\neg c \wedge \neg d) \vee (c \wedge d))))$	11	6	3
37	$(\neg a \wedge ((\neg b \wedge ((\neg c \wedge \neg d) \vee (c \wedge d)) \vee (b \wedge \neg c \wedge d)))) \vee (a \wedge \neg b \wedge \neg c \wedge d)$	13	7	3
38	$(\neg a \wedge ((\neg b \wedge ((\neg c \wedge \neg d) \vee (c \wedge d)) \vee (b \wedge \neg c \wedge d)))) \vee (a \wedge \neg b \wedge c \wedge \neg d)$	13	8	3
39	$(\neg a \wedge ((\neg b \wedge ((\neg c \wedge \neg d) \vee (c \wedge d)) \vee (b \wedge \neg c \wedge d)))) \vee (a \wedge b \wedge c \wedge \neg d)$	13	8	3
40	$((\neg c \wedge \neg d) \vee (c \wedge d)) \wedge ((a \wedge b) \vee (\neg a \wedge \neg b))$	8	8	3
41	$(a \wedge b \wedge ((\neg c \wedge d) \vee (c \wedge \neg d))) \vee (\neg a \wedge \neg b \wedge ((\neg c \wedge \neg d) \vee (c \wedge d)))$	12	7	3

distributed.

3.2 Visual Analysis

First, we will take a look at the scatter plots for concept complexity vs. number of nodes. We can see below the scatter plots containing the results for all the families, in both number of nodes and on the depth of the BDDs.

We can see in all the plots that there is a positive correlation between the number of nodes and the complexity of a subject. When looking at the BDD depths we can see smaller, but still positive correlations, in most of the families. However, for families 3[3] (3.6) we can see that the depth of the BDDs stays invariant.

In order to check whether there is an overall trend, we can look at the graphs that include all points from all the previous plots. We can see the two scatter plots of the difficulty for each of the concepts and the depths (Figure 3.8) of the corresponding BDD, as well as the number of nodes in the BDD (Figure 3.4).

We can see that there is a positive correlation between the complexity of the subject and the depth of the corresponding BDD. However, we see that the maximum depth does not exceed three, while there is only one case where the depth is zero. We can also see that depth fluctuates back and forth between two and three for concepts with complexity between four and ten.

Looking at the second plot we see that there is more correlation between the two. However, we again see the same fluctuations where the concepts on the same level of complexity have different amounts of nodes. We can see this in concept with complexity of eight especially, where the number of nodes is in the range between five and eight.

3.3 Correlation

Next, we will take a look at the results of the correlation tests. We see a moderate correlation between the depth of the BDD and the complexity of a concept ($r(39)=.67$, $p < .001$). We see a much stronger positive correlation between the number of nodes in the BDD and the complexity of the concept ($r(39)=.86$, $p < .001$).

4 Discussion

4.1 Reults

In this paper, we looked at a number of concepts and we examined whether we could derive how complex a concept is by looking at the BDD representation of the concept. We found a moderate correlation between the complexity of a concept and the depth of the BDD representation of the same concept. We also found a strong correlation between the complexity and the number of nodes in the BDD representation. Both of these results were significant.

The analysis supports the theory that there exists a connection between the BDD representation of a concept and its complexity. Although both dependent variables we looked at showed a significant positive correlation, the correlation between the number of nodes and complexity is much stronger than that of the BDD depth.

We can see a similar pattern in these results as in the Feldman (2000) paper. In some of the families, we see concepts with the same complexity value, however, they have different numbers of nodes. An example of this can be seen in the graph for families 4[4] and 3[4] (Figure 3.3). This is consistent with Feldman (2000) results, where concepts from different families with equal complexity had different proportions of correct answers.

4.2 Potential Problems

One of the potential problems is the correlation between the depth of the BDDs and the complexity of the concept. Although we did find a significant and moderate correlation between the two, we can see in Figure 3.6 and 3.7 that the depth of the BDDs does not change although the different concepts have very different complexities (ranging from 3 and 8 for family 3[3], and from 4 to 10 for family 4[3]). Furthermore, in other families, we see that the depth value does not change much for the different concepts, while the complexity of the concepts changes quite a bit. This can be seen in family 4[4] (Figure 3.7) where most of the depths are 3, with only one of them being 2 and one being 1. This pattern is visible in all of the families with depths. This could mean that the depth might not be a very useful metric in our context, since we cannot make reliable predictions about complexity through just the depth.

We can also see this pattern in the number of nodes vs. complexity of the concepts, to a lesser extent. This might mean that this measure is also not very good at predicting the complexity exactly. However, the fact that the correlation is so strong leads me to believe that it is enough to predict it with a high degree of confidence.

Another potential problem is in the assumptions of the correlation test. As previously stated this test requires that the data is normally distributed. Although the Shapiro-Wilk normality test we conducted showed normal distributions for all of the data we are using, it is quite hard to detect deviation from normality with small data sets such as the one we are using here. However given the fact

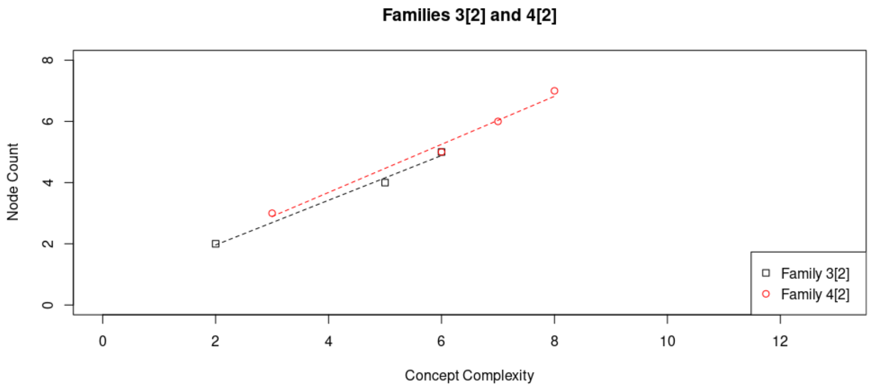

Figure 3.1: Families 3[2] and 4[2]

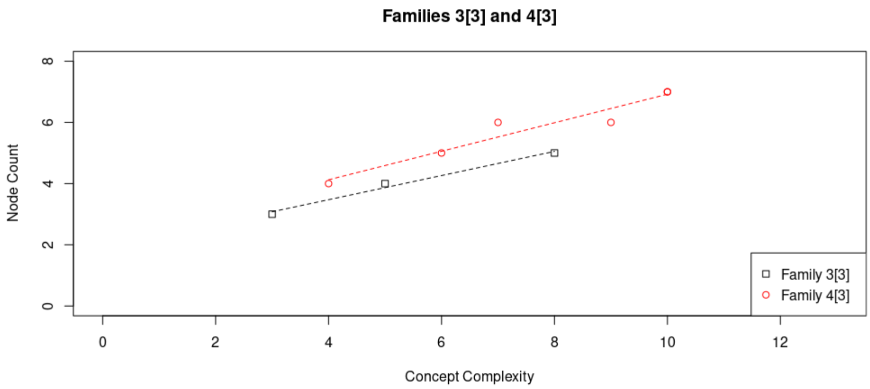

Figure 3.2: Families 3[3] and 4[3]

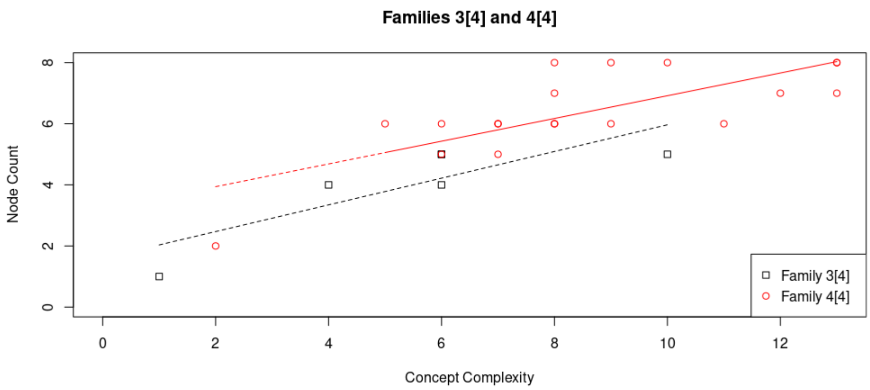

Figure 3.3: Families 3[4] and 4[4]

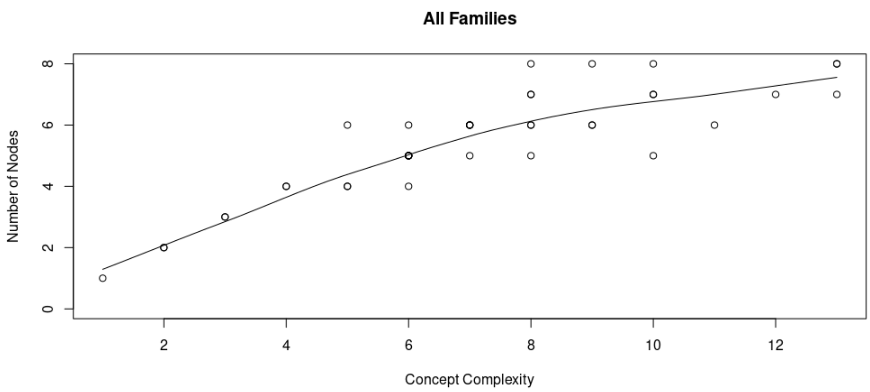

Figure 3.4: All Families

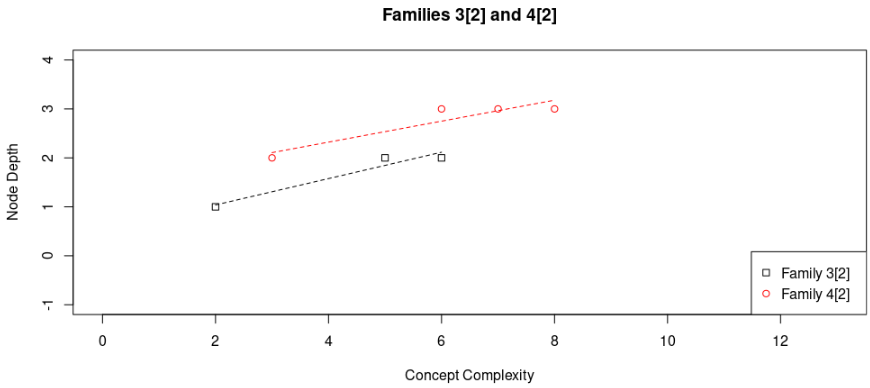

Figure 3.5: Families 3[2] and 4[2]

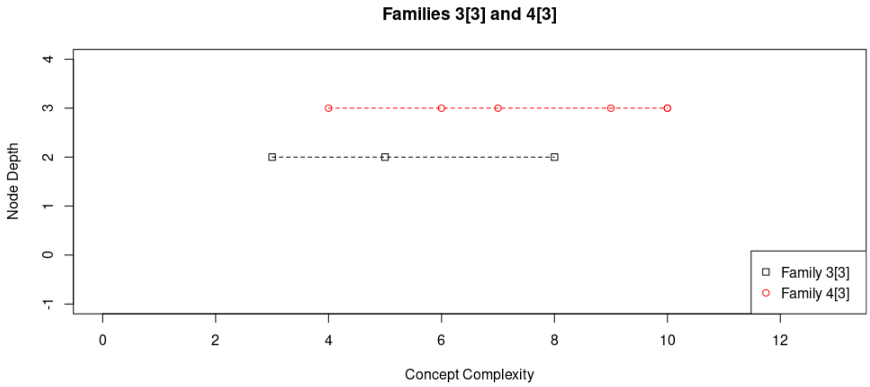

Figure 3.6: Families 3[3] and 4[3]

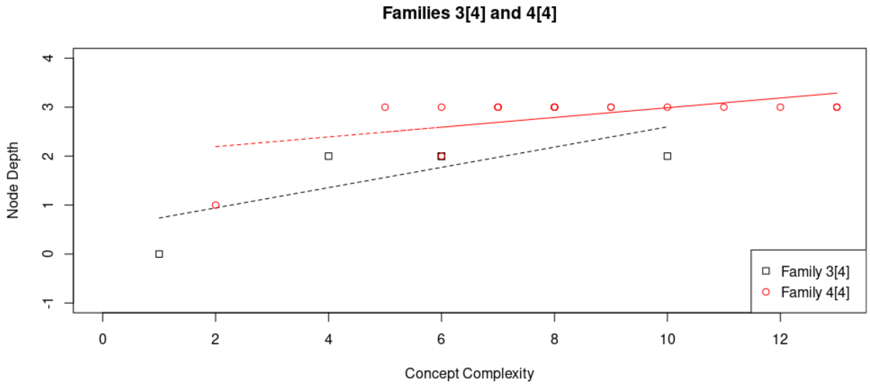

Figure 3.7: Families 3[4] and 4[4]

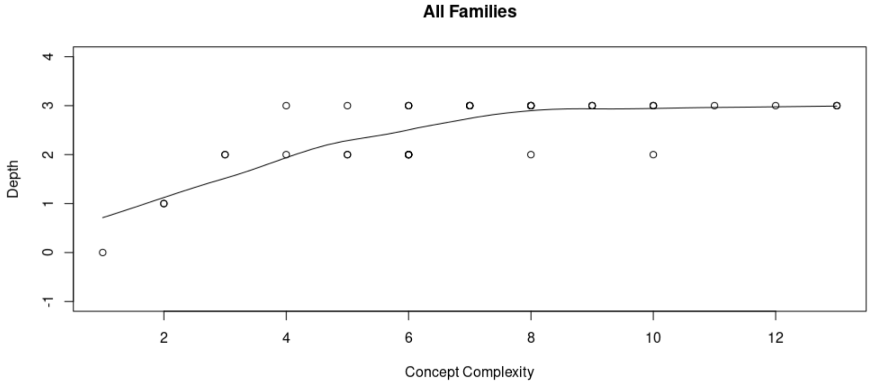

Figure 3.8: All Families

that both the subjective analysis and the statistical analysis about the distribution indicated that it is in fact normal, this should not be a problem.

4.3 Limitations and Future Work

Formulas

In this study, we only looked at the families that were included in the Feldman (2000) paper. Some of these families did not have many formulas, however adding more, higher-order families would not be a very good representation. With more formulas tested we would have higher and higher levels of boolean complexity, however this would not give us much information about formulas with small boolean complexity. As we can see in families 4[4] and 4[3], the number of higher complexity formulas is higher as compared to 3[2] and families with D < 4 and P < 3. This means that in families with D and P larger than this the complexity would increase for the most part. The low number of concepts tested here is a limitation of this study.

For future work, it is recommended that more formulas are used in the testing, especially formulas and from families with lower complexity.

Variable Ordering

Another limitation of the study was the use of only one type variant of variable ordering. Variable ordering was a subject that was outside the scope of this study. For all the concepts examined in this paper the best possible ordering was used, meaning that for a function with variables x_1, x_2 and x_3 we would order them as such: $x_1 < x_2 < x_3$. If the ordering had been $x_1 < x_3 < x_2$ then the BDD would have looked different and would have been less efficient. However, there might be a certain variable ordering that would yield a higher correlation between the dependent variables discussed above and the complexity of the concept. If such an ordering exists it would offer better predictions about the complexity of a subject as compared to the ordering used in this study.

For future work, it is recommended that a number of different variable ordering are used. This would make it possible to discover a certain variable ordering that allows us to predict the complexity of the concepts better than what we have right now if such an ordering exists.

Boolean Complexity

Lastly, we only compared our results and tested the correlation with the Boolean complexity of the propositional formulas. In the original study by Feldman (2000), and also in the study by Shepard et al. (1961) the tests were conducted on human participants and their performance was recorded in order to determine the complexity of the subjects.

For future work, it is recommended that either the original human data from the Feldman (2000) is used. Even better would be to obtain new human performance results including the list of formulas considered here and also other formulas.

5 Conclusion

In conclusion, we found a significant correlation between the Boolean complexity of a concept and the number of nodes in the BDD corresponding to that concept. We also found a significant correlation between the depth of the BDD and the Boolean complexity of the concept however this correlation was a moderate one, as opposed to the strong correlation of the number of nodes.

Another potential problem with the depth variable is that it has quite a small range as compared to the range of complexities (0-3 as opposed to 1-13). Combined with the fact that we see the same depth across a number of BDDs with complexities that vary quite a lot, we can conclude that this variable is not useful to predict concept complexity.

However, we saw that the number of nodes can in fact be used to predict the complexity of a concept with the method used here.

References

Lyle E Bourne. Knowing and using concepts. *Psychological Review*, 77:546–556, 1970. doi:10.1037/h0030000.

Randal E Bryant. Symbolic boolean manipulation with ordered binarydecision diagrams. *ACM Computing Surveys*, 24:293318, 1992. doi:10.1145/136035.136043.

Jennifer Chee. Pearson's product-moment correlation: Sample analysis. 01 2013.

John Ellson, Emden R. Gansner, Eleftherios Koutsofios, Stephen C. North, and Gordon Woodhull. *Graphviz and Dynagraph — Static and Dynamic Graph Drawing Tools*, pages 127–148. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-642-18638-7. doi:10.1007/978-3-642-18638-7_6. URL https://doi.org/10.1007/978-3-642-18638-7_6.

Jacob Feldman. Minimization of boolean complexity in human concept learning. *Nature*, 407: 630–633, 10 2000. doi:10.1038/35036586.

David Kebo. The cudd package, bdd, add tutorial and examples, 2017. URL https://www.davidkebo.com/cudd.

Lloyd K. Komatsu. Recent views of conceptual structure. *Psychological Bulletin*, 112:500–526, 11 1992. doi:10.1037/0033-2909.112.3.500.

Douglas L. Medin and Edward E. Smith. Concepts and concept formation. *Annual Review of Psychology*, 35:113–138, 01 1984. doi:10.1146/annurev.ps.35.020184.000553.

Robert M. Nosofsky, Thomas J. Palmeri, and Stephen C. McKinley. Rule-plus-exception model of classification learning. *Psychological Review*, 101:53–79, 1994. doi:10.1037/0033-295x.101.1.53.

Daniel N. Osherson and Edward E. Smith. On the adequacy of prototype theory as a theory of concepts. *Cognition*, 9:35–58, 01 1981. doi:10.1016/0010-0277(81)90013-5.

Michael I. Posner and Steven W. Keele. On the genesis of abstract ideas. *Journal of Experimental Psychology*, 77:353–363, 1968. doi:10.1037/h0025953.

Eleanor H. Rosch. Natural categories. *Cognitive Psychology*, 4:328–350, 05 1973. doi:10.1016/0010-0285(73)90017-0.

Roger N Shepard, Carl I Hovland, and Herbert M Jenkins. Learning and memorization of classifications. *Psychological Monographs: General and Applied*, 75:1–42, 1961. doi:10.1037/h0093825.

F. Somenzi. Cudd: Cu decision diagram package release 3.0.0. 2015.