

Finding Temporal Relations: Causal Bayesian Networks vs. C4.5

Kamran Karimi and Howard J. Hamilton

Department of Computer Science
University of Regina
Regina, SK
Canada S4S 0A2
{karimi,hamilton}@cs.uregina.ca

Abstract. Observing the world and finding trends and relations among the variables of interest is an important and common learning activity. In this paper we apply TETRAD, a program that uses Bayesian networks to discover causal rules, and C4.5, which creates decision trees, to the problem of discovering relations among a set of variables in the controlled environment of an Artificial Life simulator. All data in this environment are generated by a single entity over time. The rules in the domain are known, so we are able to assess the effectiveness of each method. The agent's sensings of its environment and its own actions are saved in data records over time. We first compare TETRAD and C4.5 in discovering the relations between variables in a single record. We next attempt to find temporal relations among the variables of consecutive records. Since both these programs disregard the passage of time among the records, we introduce the flattening operation as a way to span time and bring the variables of interest together in a new single record. We observe that flattening allows C4.5 to discover relations among variables over time, while it does not improve TETRAD's output.

1 Introduction

In this paper we consider the problem of discovering relations among a set of variables that represent the states of a single system as time progresses. The data are a sequence of temporally ordered records without a distinguished time variable. Our aim is identify as many cases as possible where two or more variables' values depend on each other. Knowing this would allow us to explain how the system may be working. We may also like to control some of the variables by changing other variables. We use data from a simple Artificial Life [6] domain because it allows us to verify the results, and thus compare the effectiveness of the algorithms.

Finding associations among the observed variables is considered a useful knowledge discovery activity. For example, if we observe that $(x = 5)$ is always true when $(y = 2)$, then we could predict the value of y as 2 when we see that x is 5. Alternatively, we could assume that we have the rule: **if** $\{(x = 5)\}$ **then** $(y = 2)$, and use it to set the value of y to 2 by setting the value of x to 5. Some researchers [1, 10]

have tried to find the stronger notion of causality among the observed variables. In the previous example, they may call x a *cause* of y .

In this paper we consider two approaches to the problem of finding relations among variables. TETRAD [9] is a well-known causality miner that uses Bayesian networks [3] to find causal relations. One example of the type of rules discovered by TETRAD is $x \rightarrow y$, which means that x causes y . From the examples in [1, 10], it appears that Bayesian networks discover more causal relations than actually exist in the domain. Bayesian networks find causality even in domains where the existence of causal relations itself is a matter a debate. For words in political texts, Bayesian networks find rules such as "Minister" is *caused* by "Prime" [10]. This suggests that there is a considerable amount of disagreement about the concept of causality. There are ongoing debates about the suitability of using Bayesian networks for mining causality [2, 4, 5, 11]. Here we apply TETRAD to identify relationships between variables without claiming that all of them are causal relationships.

C4.5 [8] creates decision trees that can be used to predict the value of one variable from the values of a number of other variables. A decision tree can easily be converted to a number of rules of the form **if** $\{(x = \mathbf{a}) \text{ AND } (y = \mathbf{b})\}$ **then** $(z = \mathbf{g})$. The variables x and y may be causing the value of z , or they may be associated together because of some other reason. C4.5 makes no claim about the nature of the relationship.

Both these programs ignore any temporal order among the records, while in the data that we use there does exist relations among the variables in consecutive records, in the sense that the values of some variables in a record affect the values of variables in later records. We will describe the method used to overcome this problem.

The rest of the paper is organized as follows. Section 2 describes the simple environment that we chose for testing the different methods. In Section 3 we first compare the results obtained from TETRAD and C4.5 when there is an association among the variables of a record, but no causality. After that we attempt to discover temporal relations among the records. Section 4 concludes the paper.

2 An Agent's View of its Environment

We use an Artificial Life simulator called URAL [12] to generate data for the experiments. URAL is a discrete event simulator with well known rules that govern the artificial environment. There is little ambiguity about what causes what. This helps to judge the quality of the discovered rules.

The world in URAL is made of a two dimensional board with one or more agents (called *creatures* in Artificial Life literature) living in it. An agent moves around and if it finds food, eats it. Food is produced by the simulator and placed at positions that are randomly determined at the start of each run of the simulator. There is a maximum for the number of positions that may have food at any one time, so a position that was determined as capable of having food may or may not have food at a given time. The agent can sense its position and also the presence of food at its current position. At each time-step, it randomly chooses to move from its current position to Up, Down, Left, or Right. It cannot get out of the board, or go through the obstacles that are

placed in the board by the simulator. In such cases, a move action will not change the agent's position. The agent can sense which action it takes in each situation. The aim is to learn the effects of its actions at each particular place.

URAL employs Situation Calculus [7] to build graphs with observed situations as the nodes and the actions as transition arcs between the situations. Agents use the graphs to store their observations of the world and to make plans for finding food. URAL differentiates between volatile and non-volatile properties of a situation. The x and y positions are non-volatile, and are used to distinguish among the situations. The presence of food is a volatile property, which means that the same situation can be in different states. The creature only keeps the last observed state of a situation. URAL was modified for this experiment to log each encountered situation in a file.

For our agent, time passes in discrete steps. At each time step, it takes a snapshot of its sensors and randomly decides which action it should perform. This results in records such as $\langle x \text{ position}, y \text{ position}, \text{is food here?}, \text{action} \rangle$. C4.5 treats the last variable in a record as the decision attribute, so if necessary the variables are rearranged in the log file. Figure 1 shows two example sequences of records. In Figure 1(a) the last variable is the action, while in 1(b) it is the x position. Here time passes vertically, from top to bottom.

$\langle x, y, f, a \rangle$
$\langle 1, 3, \text{false}, L \rangle$
$\langle 0, 3, \text{false}, L \rangle$
$\langle 0, 3, \text{true}, D \rangle$
$\langle 0, 4, \text{false}, U \rangle$
$\langle 0, 3, \text{true}, D \rangle$

(a)

$\langle y, f, a, x \rangle$
$\langle 3, \text{false}, L, 1 \rangle$
$\langle 3, \text{false}, L, 0 \rangle$
$\langle 3, \text{true}, D, 0 \rangle$
$\langle 4, \text{false}, U, 0 \rangle$
$\langle 3, \text{true}, D, 0 \rangle$

(b)

Fig. 1. Two example sequences of records

The agent, moving randomly around, can visit the same position more than once. Unlike the real-world data studied by many people [1, 2, 10], here we can reliably assume a temporal order among the saved records, as each observation follows the previous one in time. Considering the similarities between data gathered by an agent and the statistical observations done by people for real-world problems, it is interesting to see if we can use the same data mining techniques to extract knowledge about this environment.

The agent can move around in this very simple world, so it has a way of changing its position by performing a move action. Creating food, on the other hand, is completely beyond its power. Finding the effects of the agent's actions requires looking at more than one record at a time (two consecutive records in this environment), because in this environment the effects of an action always appears at a later time. Any algorithm that does not consider the passage of time is limiting itself in finding causal rules. So this domain contains causal relations detectable by the agents over time (the effects of moving), as well as relations that are not detectable by the agent (the place of food).

3 Experimental Results

The effectiveness of TETRAD and C4.5 at finding valid relations is assessed in two situations: within a single record and within consecutive pairs of records.

3.1 Experiment 1: Relationships within a Single Record

From the semantics of the domain, we know that no causal relationship exists within a single record. Causal relations appear across the records. However, there is an association between the x and y position of an agent and the presence of food at that position, as the simulator places food at only certain places. A position may or may not contain food at any given time. We created a log file of the first 1000 situations encountered by a single agent and used it for the experiments.

We first fed the log file to TETRAD version 3.1. In TETRAD's notation, $A \bullet \rightarrow B$ means that either A causes B , or they both have a hidden common cause, $A \bullet \bullet B$ means that A causes B or B causes A , or they both have a hidden common cause, and $A \leftrightarrow B$ means that both A and B have a hidden common cause.

TETRAD would not accept more than 8 different values for each variable, so the world was limited to an 8×8 square with no obstacles. In the log file, x and y denote the agent's position, f denotes the presence of food, and a is the performed action. The presence of food and the actions were represented by numerical values to make them compatible with what TETRAD expects as input. The results were generated by the "Build" command. It did assume the existence of latent common causes, and used the exact algorithm. TETRAD's output is shown in Table 1.

Case	Significance Level(s)	Discovered Rules	C	PC	W
1	0.0001	$y \bullet \rightarrow x, f, a \bullet \bullet x$	1	1	1
2	0.001, 0.005, 0.01	$a \bullet \bullet y, x \bullet \bullet a, x \bullet \rightarrow y, f \bullet \rightarrow y$	0	2	2
3	0.05, 0.1	$x \bullet \bullet y, f \bullet \rightarrow x, a \bullet \bullet x, f \bullet \rightarrow y, a \bullet \bullet y$	0	2	2
4	0.2	$x \bullet \bullet y, x \bullet \bullet f, x \bullet \bullet a, y \bullet \bullet f, y \bullet \bullet a, f \bullet \bullet a$	0	4	2

Table 1. The rules discovered by TETRAD.

Based on the rules enforced by URAL in the artificial environment, the desired output in TETRAD's notation would be the following relations: $x \bullet \bullet f$ (x and f are associated), $y \bullet \bullet f$ (y and f are associated), f (f has no cause and does not cause others). The relations that are not totally wrong are shown in bold. In the table the C column indicates the number of Correct rules, the PC column show the number of Partially Correct (non-conclusive) rules, and W shows the number of Wrong rules.

The appearance of food at a certain position depends on a random variable inside the URAL code, and there is no causal relation that the agent can discover, but there is an association between positions and the presence of food.

In case 1 the rule $a \bullet \rightarrow x$ correctly guesses that a may be a cause of x . However this is not conclusive in the sense that it considers it possible for a hidden common cause to exist. This is an important distinction. In the next rule f , TETRAD identifies f as something that does not cause anything, and is not caused by anything else. The

other rule, $y \bullet \rightarrow x$, is wrong because the y coordinate of a position does not determine the x coordinate. Case 2 does not go wrong in the first two rules, even though none of them is conclusive. The other two rules are wrong. Case 3 does better in finding the relationships among a and x , and a and y , but the results are still not conclusive. Case 5 finds associations among x , y and f , but then wrongly does the same for a and f too.

As seen, TETRAD draws many wrong conclusions from the data, and with the exception of one rule (f), the rest are not conclusive. Notice that here we have been generously interpreting the rules involving a as if TETRAD is aware that an action will have an effect on the *next* value, and not the current value, of x or y .

We then tried the `c4.5rules` program of the C4.5 package in the default configuration, on the same data. We assigned the presence of food as the decision attribute. C4.5rules eliminates unneeded condition attributes when creating rules. For example, if the value of x_2 is sufficient to predict the outcome regardless of the value y_2 , the generated rule will not include y_2 . We are looking for rules of the form **if** $\{(x = \mathbf{a}) \text{ AND } (y = \mathbf{b})\}$ **then** $(f = \mathbf{g})$, which means that there is a relation between the position (x and y) and the presence of food.

Table 2 shows the `c4.5rules` program's results for determining the value of f . Rules that are actually and useful for finding food are shown in bold. The rules predicting the *presence* of food correctly included both x and y . The rest of the rules deal with cases where no food was present.

Decision Attribute	Condition Attribute(s)	Number of Rules	Example	Correctness
f	x	1	If $\{(x = 5)\}$ then $(f = 0)$	Correct
f	y	2	If $\{(y = 2)\}$ then $(f = 0)$	Correct
f	a	1	If $\{(a = L)\}$ then $(f = 0)$	Wrong
f	x, y	2	If $\{(x = 2) \text{ AND } (y = 3)\}$ then $(f = 1)$	Correct

Table 2. Attributes used in rules generated by C4.5 to determine the presence of food.

C4.5 was unable to find useful rules for determining the values of x or y , as they depend on the previous position and action, which are not available. The decision tree for x , for example, wrongly included the presence of food as a condition attribute.

3.2 Experiment 2: Relationships among Consecutive Records

As mentioned, the log file consists of temporally ordered records. Neither C4.5 nor TETRAD considers the temporal order and adding a simple discrete time stamp as an attribute will not allow them to find temporal relationships. With the semantics of our example domain in mind, the most one can hope for in the previous tests is finding a correct association between the agent's position and the presence of food.

The effects of an action will not be seen until later in time. Using a preprocessing step, a *flattened log file* is created with two or more consecutive records as a single record. Flattening the sequences in Figure 1(a) and 1(b) using a *time window* of size 2 gives the sequences shown in Figure 2(a) and 2(b) respectively, where time passes horizontally from left to right and also vertically from top to bottom. Here we have renamed the variables to remove name clashes. With the exception of the first and last

records, every record appears twice, once as the second half (effect), and then as the first half (cause) of a combined record.

$\langle x_1, y_1, f_1, a_1, x_2, y_2, f_2, a_2 \rangle$
$\langle 1, 3, \text{false}, L, 0, 3, \text{false}, L \rangle$
$\langle 0, 3, \text{false}, L, 0, 3, \text{true}, D \rangle$
$\langle 0, 3, \text{true}, D, 0, 4, \text{false}, U \rangle$
$\langle 0, 4, \text{false}, U, 0, 3, \text{true}, D \rangle$

(a)

$\langle y_1, f_1, a_1, x_1, y_2, f_2, a_2, x_2 \rangle$
$\langle 3, \text{false}, L, 1, 3, \text{false}, L, 0 \rangle$
$\langle 3, \text{false}, L, 0, 3, \text{true}, D, 0 \rangle$
$\langle 3, \text{true}, D, 0, 4, \text{false}, U, 0 \rangle$
$\langle 4, \text{false}, U, 0, 3, \text{true}, D, 0 \rangle$

(b)

Fig. 2. The flattened sequences of Figures 1(a) and 1(b).

The appropriate size of the time window depends on the domain. It should be wide enough to include any cause and all its effects. If we suspect that the effects of an action will be seen in the next two records, then we may flatten Figure 1(a) to get records like: $\langle 1, 3, \text{false}, L, 0, 3, \text{false}, L, 0, 3, \text{true}, D \rangle$. The algorithms accepting the flattened records as input may not know about the passage of time, but flattening brings the causes and the effects together, and the resulting record then has the information about any changes in time.

For the next experiment, a window size of 2 was used because in the URAL domain the effects of an action are perceived by the agent in the next situation. In the combined record, x_1, y_1, f_1 and a_1 belong to the first record, and x_2, y_2, f_2 and a_2 belong to the next one. TETRAD's output for the resulting data is shown in Table 3.

Case	Significance Level(s)	Discovered Rules	C	PC	W
1	0.0001	$y_1 \bullet \rightarrow x_1, a_1 \bullet \rightarrow x_1, x_1 \bullet \bullet \bullet x_2, y_2 \bullet \rightarrow x_1,$ $a_2 \bullet \rightarrow x_1, y_1 \bullet \rightarrow x_2, y_1 \bullet \textcircled{R} y_2, a_1 \bullet \textcircled{R} x_2,$ $a_1 \bullet \leftarrow y_2, a_2 \bullet \rightarrow a_1, y_2 \bullet \rightarrow x_2, a_2 \bullet \rightarrow x_2,$ f_1, f_2	2	4	8
2	0.0005, 0.001, 0.005, 0.01	$y_1 \bullet \rightarrow x_1, x_1 \bullet \bullet \bullet a_1, x_1 \bullet \bullet \bullet x_2, x_1 \bullet \bullet \bullet y_2,$ $a_2 \bullet \rightarrow x_1, y_1 \bullet \rightarrow a_1, y_1 \bullet \rightarrow x_2, y_1 \bullet \textcircled{R} y_2,$ $a_1 \bullet \bullet \bullet x_2, a_1 \bullet \bullet \bullet y_2, a_2 \bullet \rightarrow a_1, x_2 \bullet \bullet \bullet y_2,$ $a_2 \bullet \rightarrow x_2, a_2 \bullet \rightarrow y_2, f_1, f_2$	2	4	10
3	0.1	$x_1 \bullet \rightarrow y_1, x_1 \bullet \bullet \bullet a_1, x_1 \bullet \textcircled{R} x_2, x_1 \bullet \bullet \bullet y_2,$ $x_1 \bullet \bullet \bullet a_2, f_1 \bullet \rightarrow y_1, a_1 \bullet \rightarrow y_1, y_1 \leftrightarrow x_2,$ $y_2 \bullet \rightarrow y_1, a_2 \bullet \rightarrow y_1, a_1 \bullet \textcircled{R} x_2, a_1 \bullet \bullet \bullet y_2,$ $a_1 \bullet \bullet \bullet a_2, y_2 \bullet \rightarrow x_2, f_2 \bullet \rightarrow x_2, a_2 \bullet \rightarrow x_2,$ $y_2 \bullet \bullet \bullet a_2$	0	3	14
4	0.2	$x_1 \bullet \bullet \bullet y_1, f_1 \bullet \rightarrow x_1, x_1 \leftrightarrow a_1, x_1 \leftrightarrow x_2,$ $y_2 \bullet \rightarrow x_1, a_2 \bullet \rightarrow x_1, f_1 \bullet \rightarrow y_1, y_1 \leftrightarrow a_1,$ $y_1 \leftrightarrow x_2, y_2 \bullet \rightarrow y_1, a_2 \bullet \rightarrow y_1, a_1 \bullet \bullet \bullet x_2,$ $y_2 \bullet \rightarrow a_1, f_2 \bullet \rightarrow a_1, a_2 \bullet \rightarrow a_1, y_2 \bullet \rightarrow x_2,$ $f_2 \bullet \rightarrow x_2, a_2 \bullet \rightarrow x_2, y_2 \bullet \bullet \bullet a_2$	0	1	18

Table 3. The rules discovered by TETRAD from the flattened records.

Here we are looking for the following relations: $x_1 \bullet \bullet \bullet f_1$ (x_1 and f_1 are associated), $y_1 \bullet \bullet \bullet f_1$ (y_1 and f_1 are associated), f_1, f_2 (f_1 and f_2 have no causes and do not cause

anything), $x_2 \dashv\dashv f_2$ (x_2 and f_2 are associated), $y_2 \dashv\dashv f_2$ (y_2 and f_2 are associated), $a_1 \rightarrow x_2$ (a_1 causes x_2), $a_1 \rightarrow y_1$ (a_1 causes x_1), $x_1 \rightarrow x_2$ (x_1 causes x_2), and $y_1 \rightarrow y_2$ (y_1 causes y_2). The relations that are not totally wrong are shown in bold.

Most rules discovered by TETRAD on this data are wrong. In comparison with Experiment 1, the increased number of variables in Experiment 2 has resulted in an increase in the number of discovered rules, most of which are either wrong or not conclusive.

We applied C4.5 to the same data. The desired output rules are of the following forms: **if** $\{(x_2 = \mathbf{a}) \text{ AND } (y_2 = \mathbf{b})\}$ **then** $(f_2 = \mathbf{g})$ (Association between x , y , and food), **if** $\{(x_1 = \mathbf{a}) \text{ AND } (a_1 = \mathbf{b})\}$ **then** $(x_2 = \mathbf{g})$ (predicting the next value of x), and **if** $\{(y_1 = \mathbf{a}) \text{ AND } (a_1 = \mathbf{b})\}$ **then** $(y_2 = \mathbf{g})$ (predicting the next value of y). The results produced by c4.5rules are shown in Table 4.

Decision Attribute	Condition Attribute(s)	Number of Rules	Validity
f_2	x_2	1	Correct
f_2	y_2	2	Correct
f_2	a_2	1	Wrong
f_2	x_2, y_2	2	Correct
x_2	x_1, a_1	32	Correct
y_2	y_1, a_1	32	Correct

Table 4. C4.5's results after flattening the records.

Rules that can actually be used for finding and reaching food are shown in bold. C4.5rules generated 32 correct rules for each of x_2 and y_2 . In a two-dimensional space, there are 4 possible actions and 8 distinct values for x_1 and y_1 . In this example the creature has explored all the world, and there are 32 (8×4) rules for predicting the next value of x_2 and y_2 . There were no changes in the rules for f_2 (the actually useful rules that predict the presence of food still depend on both x_2 and y_2) even though C4.5 now has more variables to choose from. This is because the current value of f_2 is not determined by any temporal relationship. Overall, C4.5 did a much better job in pruning the irrelevant attributes than TETRAD.

4 Concluding Remarks

People interested in finding relations among observed variables usually gather data from different systems at the same time. Here we used data that represented the state of a single system over time. While there were relations among the variables in each state, some interesting temporal relations existed among the variables of different states.

We applied a causality miner that uses Bayesian networks to find relations in a very simple and well-defined domain. The results were similar to the real-world problems: both correct and wrong rules were found. Bayesian causality miners need a domain expert (a more powerful causal relation discoverer) to prune the output. Flattening the records to give the algorithm more relevant data resulted in many more

irrelevant rules being discovered. We also tested C4.5 on the same data, and observed that it is very good in pruning non-relevant attributes of the records and finding temporal relations without actually claiming to be a causality discoverer. One consideration with C4.5 is that the user has to identify the decision attribute of interest.

We observed that flattening enabled C4.5 to discover new relations that it could not find otherwise. Flattening increases the number of variables in the resulting records. While creating a bigger search space for rule mining, flattening gives more information to the rule miner, and makes temporal relations explicit.

Acknowledgements

We thank Yang Xiang, Clark Glymour, and Peter Spirtes for their help. Funding was provided by a Strategic Grant from Natural Science Engineering Research Council of Canada.

References

1. Bowes, J., Neufeld, E., Greer, J. E. and Cooke, J., A Comparison of Association Rule Discovery and Bayesian Network Causal Inference Algorithms to Discover Relationships in Discrete Data, *Proceedings of the Thirteenth Canadian Artificial Intelligence Conference (AI'2000)*, Montreal, Canada, 2000.
2. Freedman, D. and Humphreys, P., *Are There Algorithms that Discover Causal Structure?*, Technical Report 514, Department of Statistics, University of California at Berkeley, 1998.
3. Heckerman, D., *A Bayesian Approach to Learning Causal Networks*, Microsoft Technical Report MSR-TR-95-04, Microsoft Corporation, May 1995.
4. Humphreys, P. and Freedman, D., The Grand Leap, *British Journal of the Philosophy of Science* 47, pp. 113-123, 1996.
5. Korb, K. B. and Wallace, C. S., In Search of Philosopher's Stone: Remarks on Humphreys and Freedman's Critique of Causal Discovery, *British Journal of the Philosophy of Science* 48, pp. 543- 553, 1997.
6. Levy, S., *Artificial Life: A Quest for a New Creation*, Pantheon Books, 1992.
7. McCarthy, J. and Hayes, P. C. Some Philosophical Problems from the Standpoint of Artificial Intelligence, *Machine Intelligence* 4, 1969.
8. Quinlan, J. R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
9. Scheines R., Spirtes P., Glymour C. and Meek C., *Tetrad II: Tools for Causal Modeling*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1994.
10. Silverstein, C., Brin, S., Motwani, R. and Ullman J., Scalable Techniques for Mining Causal Structures, *Proceedings of the 24th VLDB Conference*, pp. 594-605, New York, USA, 1998.
11. Spirtes, P. and Scheines, R., Reply to Freedman, V. McKim and S. Turner (editors), *Causality in Crisis*, University of Notre Dame Press, pp. 163-176, 1997.
12. <ftp://orion.cs.uregina.ca/pub/ural/URAL.java>