Automated Discovery of Causality and Polarity from Data

N. Tellioğlu[1,2,3] and Y. Barlas[2]

[1]nefel.tellioglu@gmail.com [2]*Industrial Engineering Department, Boğaziçi University, Istanbul, Turkey*

[3]*School of Computing and Information Systems, The University of Melbourne, Australia*

Keywords: *model conceptualization; data analysis in system dynamics; causality; polarity discovery*

ABSTRACT

In System Dynamics, like any other modeling field, the modeler tries to use data as much as possible, since model construction time and model subjectivity can be reduced by the data analysis. In this research, our focus is the use of automated data analysis in determining the polarity of causal effects. The aim is to find the polarity of the links between variables by using historical field data, under the assumption that we know all the variables influencing a given effect variable. We propose an algorithm, *discoverpolarity*, which is tested with four different datasets. The results are compared with Spearman's correlation analysis. The results show that under the assumptions of *discoverpolarity*, it outperforms Spearman correlation analysis when there is collinearity in the dataset. *Discoverpolarity* can obtain meaningful and useful results for a wide range of causal relations. However, the algorithm is sensitive to the selection of the input variables and the data sampling. The algorithm gives an opportunity to modelers to strengthen their *a priori* knowledge on link polarities. Moreover, modeler does not have to have *ceteris paribus* experimental data or information from the literature. In case of high collinearity, the algorithm may return multiple possible polarities instead of a unique solution. In such cases, the modeler must choose the most plausible of the returned polarities. In further research, we plan to make *discoverpolarity* more robust to the input parameters. After enough tests with synthetic data, the algorithm must be tested with real data before it can be used in real-life modeling.

1. INTRODUCTION

System Dynamics (SD) models are shaped by real-life experiences, scientific literature, or data in order to derive the complex structures behind dynamic problems (Sterman, 2000; Barlas, 2002). In initial stages of modeling ('conceptualization'), to decide about the direction of each causal effect, SD modelers generally use *a priori* knowledge (real-life experience or scientific literature), rather than automated data analysis. But thanks to the increasing amount of recorded data and the progress in data science tools, data have become more and more important in model construction particularly in recent years. It is desirable to ground model construction in data as much as possible, since model construction time and model subjectivity can be both reduced by the automated analysis of historical data of model variables.

In this study, we assume that we have non-ceteris paribus field data about multiple system variables influencing a given effect variable. In this case, automated polarity discovery becomes a nontrivial, sometimes impossible problem. Our research aim is to find an algorithm to determine if the links are positive or negative between variables by making use of historical field data, under the assumption that we assume/know all the cause variables that have significant influence on the effect variable (see Figure 1). We also assume that all the causal relations (functions) are either *monotonically* increasing or decreasing, which is typically an accepted best practice in SD formulations (see Sterman, 2000 for instance).

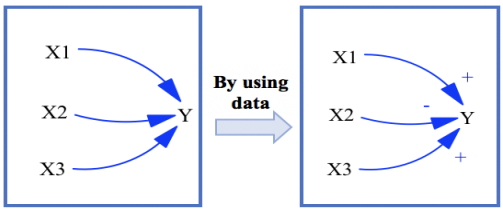

Figure 1. Discovering the Polarity of Relations.

1.1. Background

To attack the problem, the concept of *monotonicity* in causality is important. In SD, an effect variable (Y_t) is affected by variables ($X_{1,t}$, …, $X_{n,t}$) through a multiplicative, additive, or hybrid combination of nonlinear *monotonic* functions ($f_1(X_1)$,…, $f_n(X_n)$) (Sterman, 2000). Since each $f_i(X_i)$ is either nondecreasing or nonincreasing function of X_i values, the maximum value of Y must be at one of the corners of domain of the cause variables.

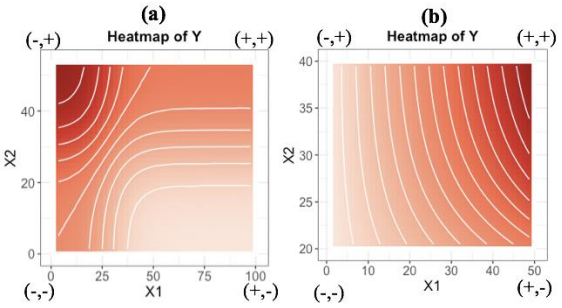

Figure 2. Heatmaps of Different Combinations of Monotonic Nonlinear Relations.

In Figure 2, some examples are given where Y variable is affected by two cause variables (X_1 and X_2) through functions of either $Y = Y_{ref} + f_1(X_1) + f_2(X_2)$ or $Y = Y_{ref} * f_1(X_1) * f_2(X_2)$. The Y values in Figure 2 are reflected as colors from red (maximum) to white (minimum). It can be seen that the maximum value of Y is achieved at one of the corners of the causal domain where the cause variables take either their minimum or maximum values. Therefore, if we know in which corner the maximum value of Y is observed, then, we can derive the *causal direction* of each effect function, since they are assumed to be monotonic. For example, in Figure 2.a, we can see that maximum value is at the corner of (-, +). This means that X_1 has a nonincreasing causal effect on Y while X_2 has a nondecreasing effect. However, in Figure 2.b, we can see that maximum value is at the corner of (+, +). This means that both X_1 and X_2 have nondecreasing effects on Y. Therefore, the question of finding the polarities of each causal relation reduces to finding the corner where Y is at its maximum (or minimum).

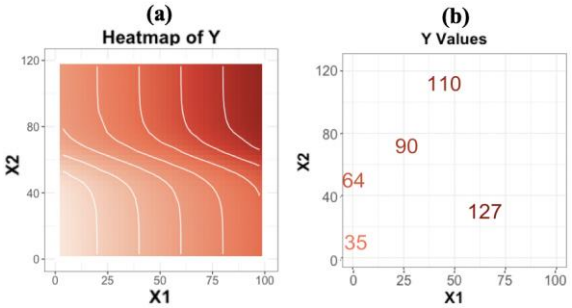

Figure 3. True Heatmap of Y Values and Limited Historical Data Available.

Unfortunately, in most field data, we do not have all the Y values that cover the causal domain. Therefore, we may not be able to directly compare the corner points to extract the polarities. Figure 3 provides such an example. We have the historical data shown in Figure 3.b, instead of having rich enough combinations of X_1, X_2, and the 'true' Y as in Figure 3.a. In this research, we try to extract the polarities of causal effects by analyzing imperfect dataset similar to Figure 3.b, which, just like real-life data, does not span the entire causal domain.

2. LITERATURE REVIEW

Correlation analysis and structural equation modeling (SEM) can be considered to extract polarities, since, both methods are commonly used to calculate strength and direction of association between variables (Tarka, 2018; Hauke & Kossowski, 2011). Even though these methods are generally applied to capture correlations between variables, they are also applied to causal relations (Kroesen, et al., 2010). When it comes to the applicability of the methods in SD causal analysis, the assumptions of both methods raise difficulties: The correlation analysis may fail when there is multicollinearity, or interdependency, among variables (Farrar, 1967). Therefore, correlation analysis would be inconvenient in SD methodology since causal variables in SD may have multicollinearity (Oliva, 2003). In case of multicollinearity, SEM can be applied by adding relationships between variables (Malhotra, 1999). However, it has different drawbacks. Hovmand and Chalise (2015) explicitly describes limitations of SEM applications in SD methodology. First of all, SEM is applicable when there is a solution to the system of differential equations. If there is strong nonlinearity in the system, then, the system may not have a solution so that SEM may not converge to a solution. Secondly, to apply SEM, system of equations should be already identified. Therefore, if not only the parameters but also some of the equations of the system are unknown, SEM cannot be used.

In recent years, the concept of monotonicity and causality in data mining algorithms has gained more importance (Hassani, *et al*., 2018). However, most of the algorithms in this literature are focused on parameter estimation for linear functions and regression analysis (Bürkner and Charpentier, 2018; Gupta, *et al.*, 2016; Pya and Wood, 2015; Hofner, *et al.*, 2016). For example, Bürkner and Charpentier (2018) developed a package in R, called *brms*, where the effects of some variables can be set as monotonic. Unfortunately, the monotonic variables must be either ordinal or discrete in the algorithm, which is not suitable for system dynamics relations. Thus, there is a gap in the related research and literature: There are no effective approaches to find the direction of nonlinear causal effects, even when the causal variables and their effective limits are known.

3. METHODOLOGY

Algorithmic discovery of the direction of each causal link in SD by only using a given dataset is a low dimensional machine learning problem, since a variable affected from more than 4-5 variables is a rare case in SD. To solve the problem, we developed an algorithm that we call ***discoverpolarity***[1]. We implemented the algorithm in R language and tested it with different datasets that satisfy the data

[1] Algorithm and the datasets are available here: https://github.com/nefeltellioglu/polaritydiscovery

assumptions stated below in Section 3.1. Data-sets are generated in R programming language. Firstly, we specify an underlying structure for each experiment and generated a dataset by using that structure. Then, we assume that we do not have the information of the link polarities. By means of known causal relations and generated dataset, we try to (re)discover the unknown link polarities with the developed algorithm (see Figure 4). Spearman (as opposed to Pearson's) correlation analysis results are used as a benchmark which is an appropriate choice for causality analysis in SD, because: Firstly, Spearman correlation analysis is non-parametric which means that there is no assumption for underlying distributions of variables. Secondly, just like in SD, it assumes monotonic relationships among variables.

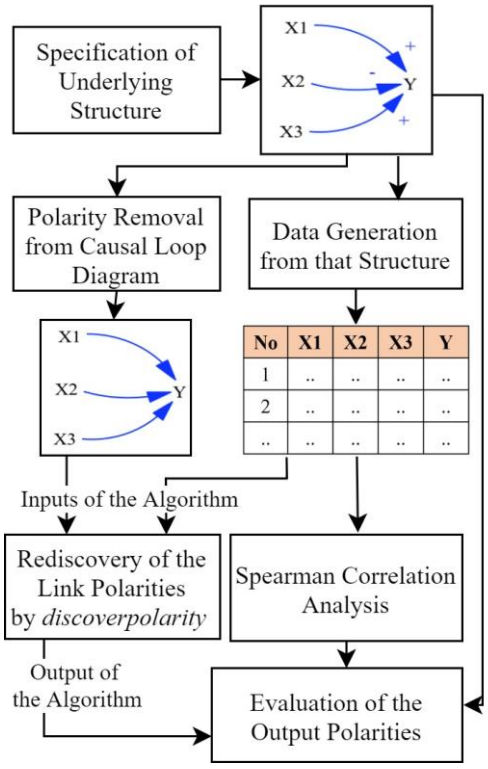

Figure 4. Research Design of Discovering Link Polarities.

3.1. Assumptions

We assume that we have real-life (non-experimental) dynamic data about multiple system variables. However, simulation-generated 'synthetic' data is used instead of actual real-life data, to be able to evaluate the validity of the results obtained from our algorithm. All data are generated by R programing language, using some assumed 'true' causal formulations between Y and Xi.

Secondly, it is assumed that the scope of the variables, the boundary of the system, is well-chosen for the problem. Therefore, when one applies the method, she does not have to consider the possibility of a non-included variable affecting the system significantly. Moreover, it is assumed that data is noise-eliminated and has no missing values.

In addition, we assume that we know the range of cause variables in which $dfi(Xi)/dX_i$ is significantly different than 0. Therefore, we can separate the range of X_i where $f_i(X_i)$ is increasing (decreasing) with respect to X_i from the X_i domain where $f_i(X_i)$ is a (approximately) constant function. For example, a significant change in cause variable X_i, which has an S-shaped causal function, may not create any impact on the effect variable after some maximum and minimum limit of X_i, since $dfi(Xi)/dX_i$ becomes zero. To apply the following method, we need to know the ranges of cause variables where the variables significantly affect Y variable. For some cause variables, $dfi(Xi)/dX_i$ may be always different than zero such as linear or certain exponential functions. (See Figure 5.a). In such cases, we do not have to consider the ranges where $dfi(Xi)/dX_i$ is almost zero and where it is significantly different than zero.

In Figure 5, we see the effect of X_1 on Y, the effect of X_2 on Y, and heatmap of Y in the causal variables domain. In this example, Y has an additive causal function: $Y = Y_{ref} + f_1(X_1) + f_2(X_2)$. As we see from Figure 5.b, X_2 variable does not affect the Y variable significantly out of the red range of $f_2(X_2)$. The limits of the red range is shown as black limits in Figure 5.d.

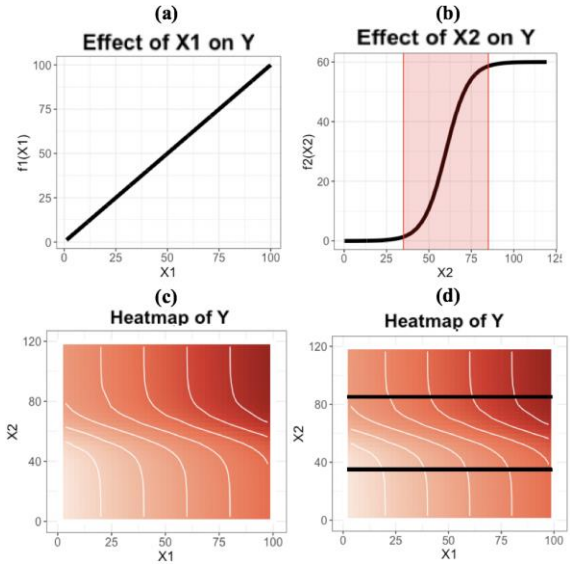

Figure 5. Effect Functions of Cause Variables on Y and the Heatmap of Y.

Therefore, in Figure 5.d, any change in X_2 does not affect Y variable when X_2 stays out of the red range. We can see that a vertical movement in out-of-red-range does not create any color (significant Y) change in Figure 5.d. However, the case is different for variable X_1. We only observe a change in Y resulting from a change in X_1 out of the black limits in Figure 5.d. Since X_1 variable linearly affects Y variable, there are no limits in the heatmap for X_1. Therefore, a significant change in X_1, a horizontal movement in Figure 5.d, always results in a change in Y variable, ceteris paribus.

4. ALGORITHM

4.1. The Basic (First) Phase of the Algorithm

Our algorithm works with proof by contradiction. It compares changes in Y while moving among observed data points in the causal domain. Then, it eliminates the corner points that cannot have the maximum value of Y. In other words, the algorithm takes the "causal variable" and respective "effect variable" values as inputs with some other parameters. Then, it compares the differences in causal variables and checks the signs of each difference. Finally, it eliminates the impossible cases from *all the possible polarities list* (names of each corner in the causal domain) which has 2 to the power of the 'number of causal variables' possibilities. *All possible polarities list* has 4 values when there are two cause variables: (+, +), (+, -), (-, +), (-, -). The process steps are given in Figure 6.

The basic process of the algorithm eliminates the corner with the same polarity of the causal variable movement while Y value is decreasing, and it eliminates the corner with the opposite polarities of the causal variable movement while Y value is increasing. Therefore, we analyze all of the paired combinations of the data points we have. In total, we check the sign of changes in n*(n-1)/2 points for a dataset composed of n points. Let's assume that we have two cause and one effect variables (X1, X2; Y). When we subtract two observed data from each other we obtain following changes in variables: (dX1, dX2, dY) = (+, +, +). This means that Y increased when X1 and X2 increased. Therefore, we can conclude that (-, -) cannot have the maximum value of Y. In other words, when we observe an increase in Y when X1 and X2 are increased, then, we can say that X1 and X2 cannot both have a decreasing effect on Y variable. Hence, we eliminate (-, -) from all the possible polarities list. In generic form, the algorithm eliminates all impossible polarities from all the possible polarities list after getting the observed change signs in differences. As the geometric interpretation, what we do is to compare (-, -) corner and (+, +) corner. It is seen that when we move from the corner (-, -) to corner (+, +), we observe an increase in Y. Therefore, corner (-, -) should be eliminated. However, by only looking at this observation, we cannot comment on the corner points (+, -) and (-, +). More data points are needed to compare the corners (+, +), (+, -) and (-, +) after elimination of (-, -).

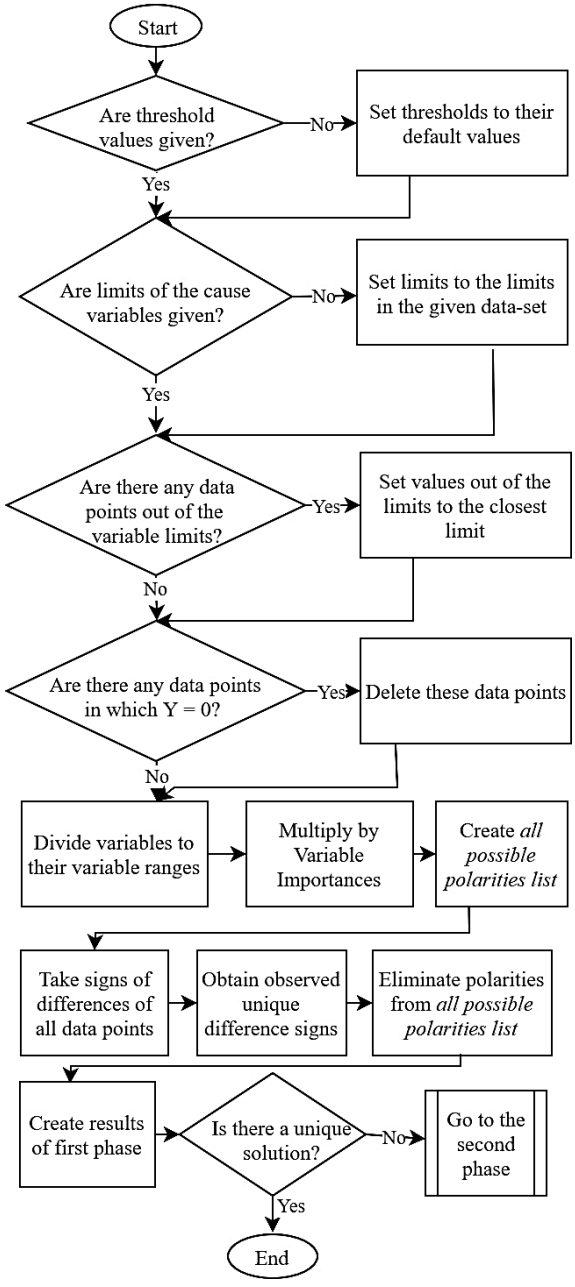

Figure 6. The Flow Chart of the First Phase of the Algorithm.

The algorithm also handles "no change" in cause (1) and effect variables (2). For no change in cause variables (1), algorithm applies the following procedure: When there is no change in some of the causal variables, the algorithm only considers the changes in the other causal variables. For instance, if we observe changes in the cause variables as (dX1, dX2, dY) = (+, 0, +), the algorithm eliminates 2

corners: (f1(X1), f2(X2))= (-, -) and (f1(X1), f2(X2) = (-, +).

When we observe no change in the Y variable while a causal variable significantly changes (2), the algorithm eliminates both corners with the same and the opposite polarities of the causal variable movement. For example, if we observe changes in the cause variables as (dX1, dX2, dX3) = (+, +, +) while d(Y)= (0), the algorithm eliminates 2 corners: (f1(X1), f2(X2), f3(X3)) = (+, +, +) and (f1(X1), f2(X2), f3(X3)) = (-, -, -).

When a value of a cause variable which is out of its range in which (dfi(Xi))/(dXi) is significantly different than 0, is observed, the observed value is set as the closest limit of that variable. Therefore, a difference between two points out of the effective range is considered as zero even if the variable changes significantly out of its range. In addition, the points where the effect variable is zero are eliminated to prevent a misleading conclusion that may result from a multiplicative causal function.

4.2. Second Phase of the Algorithm

In some collected data, there may not be enough information to compare and eliminate many corners. Therefore, we end up with multiple cases in a dataset that have poor information. In other words, because of multicollinearity or insufficient data, we may end up with multiple possibilities instead of a unique solution.

When we end up with multiple options from the first phase, the algorithm moves into the second phase. At this point, the algorithm has a default assumption: in the range of the given limits of the variables, the maximum ceteris paribus changes created by all X_i's on Y are (approximately) the same. In other words, influence importances of X_i's on Y are the same. However, if the modeler knows the maximum ceteris paribus impact of each variable which is significantly different than each other, then she must give these impact levels (influence importances) as an input, *varImp*. The flow chart of the second phase given in Figure 7.

In the second phase, the algorithm takes the increasing or decreasing amounts of change of cause and effect variables into account. We already know that a significant change of a cause variable must produce a change (ceteris paribus) in the effect variable. Therefore, when a cause variable X_i does not change much but the other cause variables $X_{j,\ j\neq i}$ change significantly, then we can say that the change in the effect variable results from $X_{j,\ j\neq i}$.

In this step, algorithm subtracts the minimum limit of the variable from the variable value and then divides the variables by their given limits (the limit of the effect variable is its maximum value). By doing so, it obtains relative changes over a range of (0, 1). Then, if variable importances are different, the algorithm multiplies relative importances (minimum is set as 1) with the difference values.

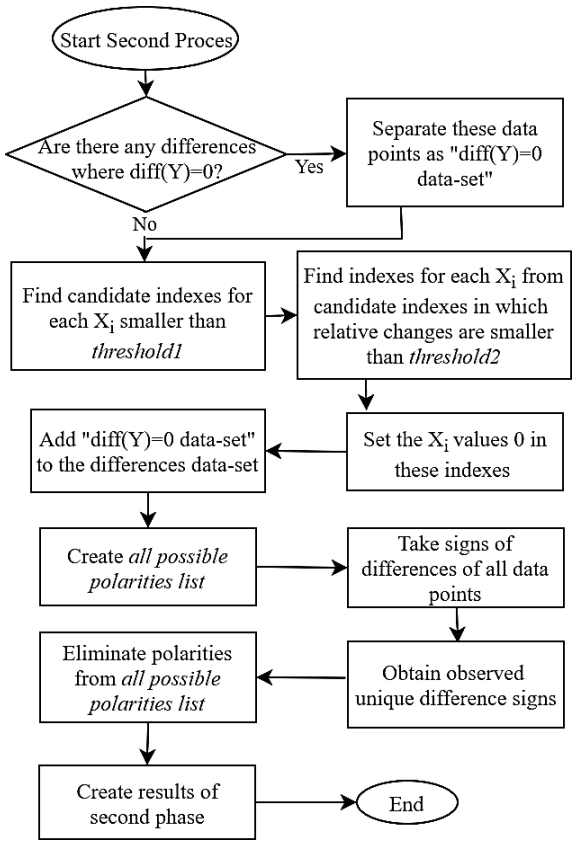

Figure 7. The Flow Chart of the Second Phase.

In the new causal domain, algorithm takes the absolute of the differences of each data point to obtain the relative absolute difference values for each causal variable. Then, the algorithm searches for points close to each other with respect to each cause variable. The points that have relative difference values smaller than threshold1 (default value is set as 0.15) are considered as 'close' according to a cause variable and are considered as *first candidate indexes*. Then, for each *first candidate index*, the algorithm checks the absolute difference values of other cause variables. If the ratio of "*the absolute difference value of candidate cause variable*/*sum of the absolute difference value of all causal variables in the candidate index*" is smaller than threshold2 (default value is set as threshold1), then the algorithm sets the difference value of the candidate cause variable at that point to zero. In other words, it assumes that the change in Y results from the other cause variables. By doing so, algorithm eliminates more corners which do not include the maximum Y value.

4.3. Inputs and Outputs of the Algorithm

Inputs of the algorithm and their definitions are given in Table 1. When the algorithm is applied, the modeler must decide on the threshold values used in the algorithm. This is the trickiest part of the algorithm since the proper values of the thresholds may differ from one dataset to another.

Table 1. Inputs of the Algorithm.

Name	Description
causes	Cause variables as data.frame()
effect	Corresponding effect variable.
threshold1	The threshold to select first candidate indexes for each cause variable to set them to zero. The default value is 15%.
threshold2	The threshold to select indexes from the first candidate indexes list to set their value as 0. The default threshold2 is the value of 15%
threshold3	The threshold to select indexes for the effect variable to set the effect variable as zero. The default value is set as the value of threshold1.
varImp	Used in the second phase. varImp of all the variables is 1 as default.
limits	The range of cause variables where the variables significantly affect Y.

Even though the algorithm has default values for the thresholds, for some datasets, the algorithm may eliminate all the possible polarities including the true polarity when default threshold values are used. This may happen if the threshold values that are used in the algorithm are too high for these datasets. In such cases, the modeler must decrease the threshold values to obtain a reasonable result. When the algorithm returns multiple results, the modeler may prefer to increase the threshold values. However, the usage of thresholds higher than 0.15 is not recommended since it can cause the elimination of true polarity because of high nonlinearity in the effect functions.

If the first phase of the algorithm finds a unique solution, it returns the solution (All1) and the eliminated possibilities table (EliminatedOptions1) with the corresponding number of differences used to eliminate the corners (NoofObservedDifferences). The idea behind returning the eliminated possibilities is to show how many point differences we had to decide that this corner must be eliminated. In some cases, the researcher may want to check the differences and the points used to get these differences when a possibility is eliminated with very small evidences (number of differences).

If the algorithm goes into the second phase, it returns results of both first and second phases (All1, EliminatedOptions1, All2, EliminatedOptions2). Therefore, the modeler can see the results for before and after applying the second phase. The solution (All2), sometimes multiple ones, and eliminated possibilities table (EliminatedOptions2) are given as the outputs of the second phases just like the first one.

5. EXAMPLES AND RESULTS

The algorithm is tested, and the results are reported for four different datasets. First dataset has two cause variables while second and third datasets have four cause variables.

5.1. Dataset 1: Multiplicative Formulation with Two Cause Variables

The first dataset has Y variable with the underlying function of $Y = Y_{ref} * f_1(X_1) * f_2(X_2)$. Each effect function and the heatmap of Y in the causal domain is given in the Figure 8.a-c. The dataset that we have with 26 samples is shown in Figure 8.d. The correlation matrix of dataset 1 is given in Table 2.

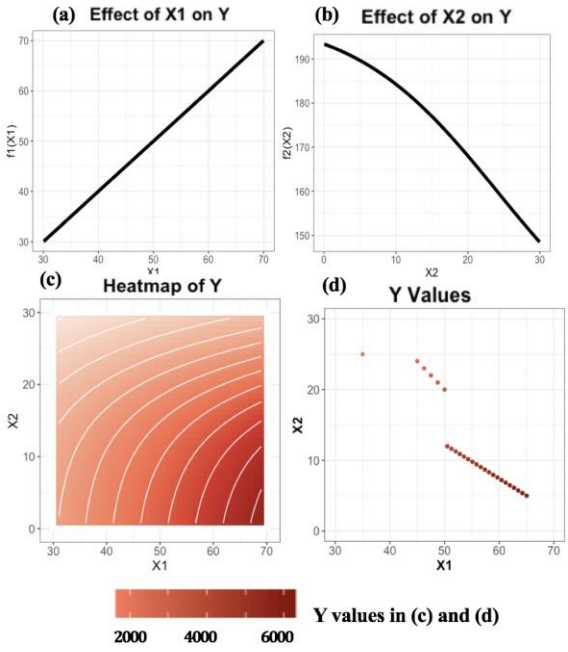

Figure 8. Underlying Functions and the Dataset 1.

Table 2. Spearman Correlation Matrix of Dataset 1.

	x1	x2	y
x1	1.0000	-1.0000	1.0000
x2	-1.0000	1.0000	-1.0000
y	1.0000	-1.0000	1.0000

In this analysis, variable importances are given as one (default value of varImp) and no variable limit is specified. Moreover, the default threshold values are used. When the algorithm is applied to dataset 1, from the first phase, the algorithm only eliminates the corner of (-, +). In the second phase, the algorithm is able to eliminate two other corners: (-, -) and (+, +). Therefore, the only left option is (+, -) which means that X_1 has a nondecreasing effect while X_2 has a nonincreasing effect on Y variable. From the heatmap in Figure 8.c, it can be seen that this is the correct conclusion.

Spearman correlation analysis cannot produce any result for dataset 1 because of perfect multicollinearity. Because of the perfect correlation of the ranks of cause variables, variance-covariance matrix of the rank values has a determinant of zero.

5.2. Dataset 2: Multiplicative Formulation with Three Cause Variables

The second dataset has Y variable with the underlying function of $Y = Y_{ref} * f_1(X_1) * f_2(X_2) * f_3(X_3)$. Effects of causal variables on Y are given in the Figure 9. From the effect functions, it is observed that X1 and X2 have positive while X3 has negative effect on Y variable. Therefore, the true corner where the maximum Y variable is located is (+, +, -). The dataset that is used to discover the causal polarities has 100 samples as it is shown in Figure 10. The correlation matrix of dataset 2 is given in Table 3.

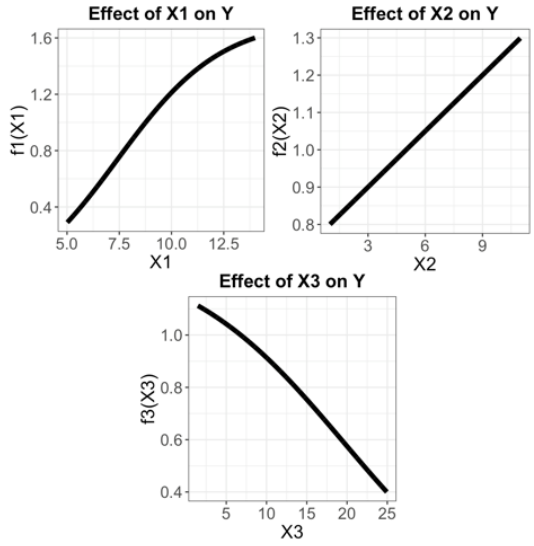

Figure 9. Underlying Functions of the Dataset 2.

In this analysis, variable importances are given as one (default value of varImp); only the upper limit of X1 variable is set as 12.5; and the threshold values are set as 0.05. When the algorithm is applied to dataset 2, from the first phase, the algorithm eliminates six corners out of eight corners. In the second phase, the algorithm eliminates one more corner. Therefore, the only left option is (+, +, -). From the effect functions in Figure 9, it can be seen that this is the correct conclusion. It is critical to mention that threshold values are chosen as 0.05 for this dataset. When the default value of 0.15 is used, the algorithm eliminates all the corners which means that the default threshold is a very high value for this dataset.

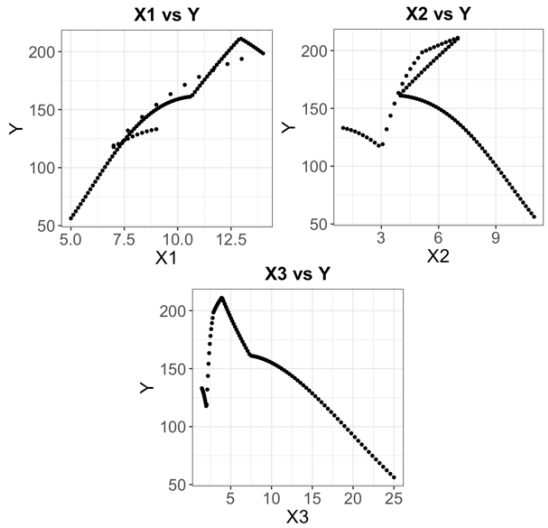

Figure 10. Cross-sectional Graphs of the Dataset 2.

Through partial spearman correlation, we obtain partial correlation coefficients as $(\rho_{X1}, \rho_{X2}, \rho_{X3}) = (0.99, -0.37, -0.61)$ with respective P-values which are close to zero. The Spearman correlation coefficient does not return the true direction of causal relations. Therefore, our algorithm gives correct results where the correlation coefficient fails to return the correct direction of causal relations.

Table 3. Spearman Correlation Matrix of Dataset 2.

	x1	x2	x3	y
x1	1.0000	-0.4137	-0.6037	0.9894
x2	-0.4137	1.0000	0.8291	-0.3769
x3	-0.6037	0.8291	1.0000	-0.5843
y	0.9894	-0.3769	-0.5843	1.0000

5.3. Dataset 3: Additive Formulation with Four Cause Variables

The third dataset has Y variable with the underlying function of $Y = Y_{ref} + f_1(X_1) + f_2(X_2) + f_3(X_3) + f_4(X_4)$. Effects of causal variables on Y are given in the Figure 11. From the effect functions, it is observed that X_1, X_3, and X_4 have positive while X_2 has negative effect on Y variable. Therefore, the true corner where the maximum Y variable is located is (+, -, +, +). The dataset that is used to discover the causal polarities has 100 samples as it is shown in

Figure 12. The correlation matrix of dataset 3 is given in Table 4.

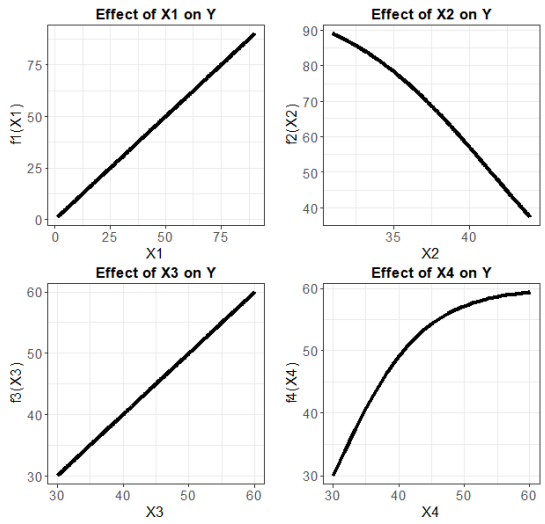

Figure 11. Underlying Functions of the Dataset 3.

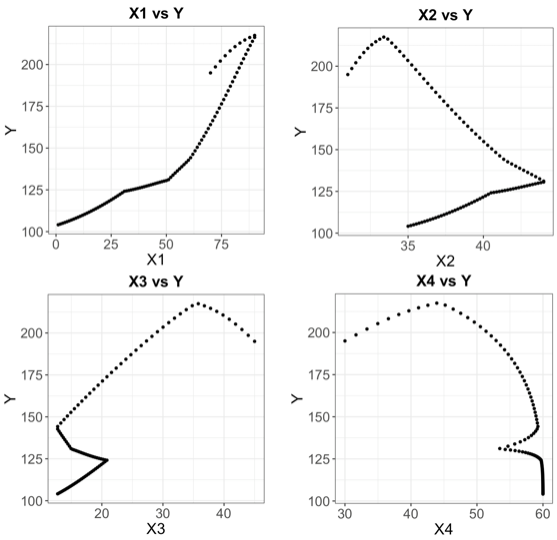

Figure 12. Cross-sectional Graphs of the Dataset 3.

In this analysis, variable importances are given as one (default value of varImp) and only the upper limit of X4 variable is specified (as 55). When the algorithm is applied to dataset 3, from the first phase, the algorithm eliminates nine corners out of sixteen corners. In the second phase, the algorithm eliminates eight other corners. Therefore, the only left option is (+, -, +, +). From the effect functions in Figure 11, it can be seen that this is the correct conclusion. It is critical to mention that threshold values are chosen as 0.05 for this dataset. When the default value of 0.15 is used, the algorithm eliminates all the corners which means that the default threshold is a very high value for this dataset.

Through partial spearman correlation, we obtain partial correlation coefficients as (ρ_{X1}, ρ_{X2}, ρ_{X3}, ρ_{X4}) = $(0.99, -0.25, 0.63, -0.91)$ with respective P-values which are close to zero. The Spearman correlation coefficient does not return the true direction of causal relations. Therefore, our algorithm gives correct results where the correlation coefficient fails to return the correct direction of causal relations.

Table 4. Spearman Correlation Matrix of Dataset 3.

	x1	x2	x3	x4	y
x1	1.0000	-0.2552	0.5985	-0.9159	0.9949
x2	-0.2552	1.0000	-0.5132	0.1710	-0.2736
x3	0.5985	-0.5132	1.0000	-0.6195	0.6120
x4	-0.9159	0.1710	-0.6195	1.0000	-0.9308
y	0.9949	-0.2736	0.6120	-0.9308	1.0000

5.4. Dataset 4: Additive Formulation with Four Cause Variables

Let us consider another dataset where Y variable has the function of $Y = Y_{ref} + f1(X1) + f2(X2) + f3(X3) + f4(X4)$. Effects of causal variables on Y are given in the Figure 13. From the effect functions, it is observed that X_1, X_2, and X_3 have negative while X_4 has positive effect on Y variable. Therefore, the true corner where the maximum Y variable is located is (-, -, -, +). The dataset that is used to discover the causal polarities has 100 samples as it is shown in Figure 14. The correlation matrix of dataset 4 is given in Table 5.

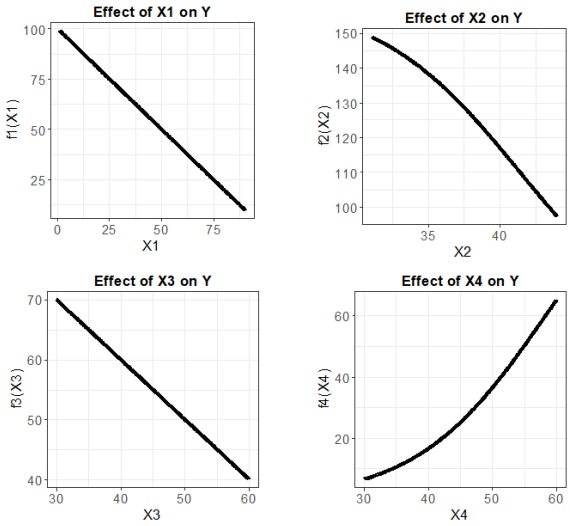

Figure 13. Effect Functions of Cause Variables on Y in Dataset 4.

In this analysis, variable importances are given as one (default value of varImp); no variable limit is specified; and the default threshold values are used. When the algorithm is applied to dataset 4, from the

first phase, the algorithm eliminates nine corners. In the second phase, the algorithm eliminates four other corners. Therefore, three corners are left as the possible polarities: (-, -, -, -), (-, -, -, +), and (-, +, +, +). From the effect functions in Figure 13, it can be seen that the correct signs are (-, -, -, +) which is one of the three left options. The default threshold values are used.

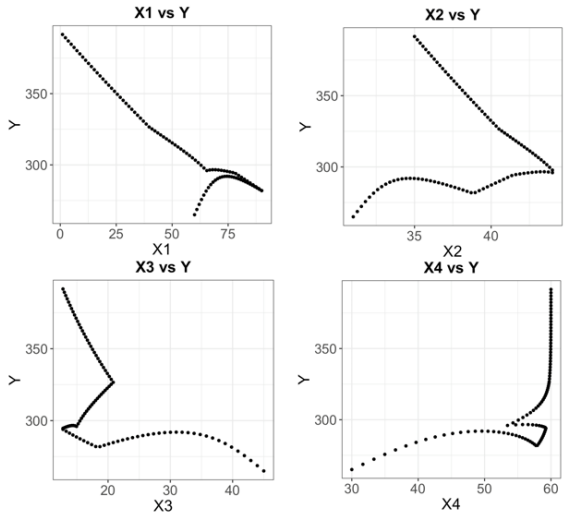

Figure 14. Cross-sectional Graphs of the Dataset 4.

Through partial spearman correlation, we obtain correlation coefficients as $(\rho_{X1}, \rho_{X2}, \rho_{X3}, \rho_{X4}) = (0.89, -0.18, -0.54, 0.83)$ with respective P-values which are close to zero. The Spearman correlation coefficient does not return the true direction of causal relations. Even though our algorithm gives multiple results, the true polarity is at least one of its results. When the number of cause variables increases, our algorithm tends to return more possibilities since it becomes unlikely for data to cover all the causal domain. However, when the number of cause variables increases, the Spearman correlation analysis tends to return wrong directions because of the strong linearity assumption among the ranked values of the variables.

Table 5. Spearman Correlation Matrix of Dataset 4.

	x1	x2	x3	x4	y
x1	1.0000	0.0709	0.2626	-0.6982	-0.8863
x2	0.0709	1.0000	-0.5132	0.1710	0.1769
x3	0.2626	-0.5132	1.0000	-0.6195	-0.5201
x4	-0.6982	0.1710	-0.6195	1.0000	0.8579
y	-0.8863	0.1769	-0.5201	0.8579	1.0000

Collinearity spectrum of datasets has two end points: "fully randomly selected" dataset and "perfectly correlated" dataset. Real-life datasets are somewhere in between, perhaps closer to "perfectly correlated" end point. In four datasets that have high collinearities, our algorithm results are compared with Spearman correlation analysis.

For the simplest multiplicative dataset with two cause variables, our algorithm returns the true causal directions whereas Spearman correlation analysis fails to return any conclusion because of multicollinearity among cause variables. In the second case with three multiplicative effects, *discoverpolarity* returns the true polarities whereas Spearman correlation analysis returns wrong causal polarities. In the third dataset with four additive effects, our algorithm returns the true directions whereas Spearman correlation analysis returns wrong causal directions. Finally, in the most difficult fourth case with four additive effects, our algorithm is able to reduce the possible polarity set to three, one of which corresponds to correct polarities. Correlation analysis on the other hand returns wrong causalities in this case.

6. CONCLUSION

It is important to use data as much as possible in model construction, since model construction time and model subjectivity can be both be reduced by formal analysis of historical data of model variables. The possibility of automated data analysis in the process of determining the polarity of the causal effects is one of the promising research areas in that context.

In this study, it is assumed that we have real-life (non-experimental) dynamic data about multiple system variables influencing a given effect variable. With multiple variables, automated polarity discovery becomes a nontrivial, sometimes impossible problem. Our research aim is to discover the polarity of the links between variables by using historical field data under the assumption that we know all the cause variables that significantly affect the given effect variable. We propose an algorithm called *discoverpolarity*, code it in R language and test it with different synthetic datasets. The test results are compared with Spearman's correlation analysis.

The results show that under the assumptions of *discoverpolarity*, it outperforms Spearman correlation analysis when there is collinearity in the dataset. *Discoverpolarity* can obtain meaningful and useful results for various underlying causal relations. However, the algorithm is sensitive to the selection of the input variables and the dataset sampling. The algorithm gives an opportunity to modelers to strengthen their *a priori* knowledge on link polarities when there is a lack of *ceteris paribus* information or data. Moreover, modeler does not have to have *ceteris paribus* experimental data or information from the literature.

It is important to note that the algorithm may return multiple possible polarities instead of a unique solution when the data only consists of all perfectly correlated data points. In such cases, the modeler must choose the most plausible of the returned polarities. In addition, modeler may need to check the link polarities that are eliminated in the second phase of algorithm if the number of the observed differences to support these eliminations are only a few. Another limitation of *discoverpolarity* is that the modeler must decide on the threshold values used in the algorithm. Even though *discoverpolarity* has default values for the thresholds, for some datasets, the default may eliminate all the possible polarities including the true polarity. This may happen if the thresholds used in *discoverpolarity* are high relative to the nature of the data. In such cases, the modeler must decrease the threshold values to obtain a reasonable result. On the other hand, if *discoverpolarity* returns multiple possible polarities, the modeler may prefer to increase the threshold values. However, usage of too high thresholds can cause the elimination of true polarity because of high non-linearity in the relations.

As further research, we plan to focus on decreasing the sensitivity of *discoverpolarity* to its threshold values and other input parameters to make it more robust. After enough tests with synthetic data, we also plan to test the algorithm with real data before it can be used in real-life modeling. Finally, not only using the signs of differences but also the magnitudes of these differences, one may be able to estimate the mathematical forms (additive, multiplicative, or hybrid) of the causal formulations. Even if fully automated discovery of causality from field data may be utopic, assisting partially the modeler's mental/informal methods in deciding on causal formulations can be a realistic and useful expectation in the near future.

REFERENCES

Barlas, Y., (2002), System Dynamics: Systemic Feedback Modeling for Policy Analysis, *In Knowledge for Sustainable Development—An Insight into the Encyclopedia of Life Support Systems, UNESCO-EOLSS: Paris and Oxford*, pp. 1131–1175.

Bürkner, P. C., & Charpentier, E. (2018). Monotonic Effects: A Principled Approach for Including Ordinal Predictors in Regression Models.

Farrar, D. E., Glauber, R. R., (1967), Multicollinearity in Regression Analysis: The Problem Revisited., *The Review of Economics and Statistics*, Volume (49), p. 92.

Gupta, M., Cotter, A., Pfeifer, J., Voevodski, K., Canini, K., Mangylov, A., Moczydlowski, W., van Esbroeck, A., (2016), Monotonic Calibrated Interpolated Look-Up Tables, *Journal of Machine Learning Research*, Volume (17), pp 1-47.

Hassani, H., Huang, X. and Ghodsi, M., (2018), Big Data and Causality, *Annuals of Data Science*, Volume (5-2), pp.133-156.

Hauke, J. and Kossowski, T., (2011), Comparison of Values of Pearson's and Spearman's Correlation Coefficients on the Same Sets of Data, *Quaestiones geographicae*, Volume (30-2), pp.87-93.

Hofner, B., Kneib, T. and Hothorn, T., (2016), A unified framework of constrained regression, *Statistics and Computing*, Volume (26-1-2), pp.1-14.

Hovmand, P. S., Chalise, N., (2015), Simultaneous Linear Estimation Using Structural Equation Modeling, *Analytical Methods for Dynamic Modelers*, pp. 71-93, MIT Press.

Kroesen, M., Molin, E.J. and van Wee, B., (2010) Determining the Direction of Causality Between Psychological Factors and Aircraft Noise Annoyance, *Noise and Health*, Volume (12-46), p.17.

Malhotra, N. K., M. Peterson, S. B. Kleiser, (1999), Marketing Research: A State-of-the-art Review and Directions for the Twenty-First Century, *Journal of the Academy of Marketing Science*.

Oliva, R., (2003), Model Calibration as a Testing Strategy for System Dynamics Models, *European Journal of Operational Research*, Volume (151), pp. 552–568.

Pya, N. and Wood, S.N., (2015), Shape Constrained Additive Models, *Statistics and Computing*, Volume (25-3), pp.543-559.

Sterman, J.D., (2000), Business Dynamics: Systems Thinking and Modeling for A Complex World (No. HD30. 2 S7835 2000).

Tarka, P., (2018), An Overview of Structural Equation Modeling: Its Beginnings, Historical Development, Usefulness and Controversies in The Social Sciences, *Quality & Quantity*, Volume (52-1), pp.313-354.