Mining for Causal Regularities

Thomas Bidinger*, Hannah Buzard*, James Hearne*, Amber Meinke*, and Steven Tanner*
Western Washington University, Bellingham, Washington 98225, USA

Abstract

This paper reports on an algorithmic exploration of the theory of causal regularity based on Mackie's theory of causes as MINUS conditions, i.e., minimal insufficient but necessary member of a set of conditions that, though unnecessary, are sufficient for the effect. We describe the algorithm to extract causal hypotheses according to this model and the results of its application to a number of real-world data sets. Results suggest further promising applications, modifications and extensions that might derive further insights of a dataset.

Key Words: Causal regularity, data mining, INUS condition, MINUS condition, Mill's methods.

1 Introduction

Of the several established approaches to the notion of causality, the regularity view is the oldest. It was introduced by Hume in the 18th century, elaborated upon by Mill in the 19th century and finds its most detailed articulation in Mackie in the 20th century. In this view, causes are to be identified as conditions or events that are uniformly accompanied or followed by some effect. Importantly, in this view, no other intrinsic relation between cause and effect is assumed other than regularity. This theory of causes invites the possibility of a search for causes by appeal to strict pattern matching, independent of statistical or probabilistic considerations. What is reported here is one approach to realizing this conception of causes and their identification.

2 Approach

In the more recent formulation by Mackie, the causal antecedents of an effect are complex configurations of facts. To motivate this view, note that a match might flare because it is struck on an abrasive surface in the right conditions – absence of moisture and presence of oxygen – or it might flare because it is heated to a flash point under similar conditions, or placed in proximity to another flame. In Mackie's formulation, a cause is what he dubs an INUS condition, an insufficient but necessary member of a set of conditions which, though unnecessary, are sufficient for the effect. Formally, this means that the search for causes is equivalent to searching for valid implications whose right-hand side is the effect and whose left-hand side is a disjunctive normal form expressing configurations of conditions: There is no a priori restriction on the number of elements in each of the conjuncts nor any restriction as to their number. Indeed, there are several other variations on this idea, as well as extra constraints discussed below. A further constraint, not originally articulated by Mackie, is that the conjunctions participating in an INUS be minimal; that is, they should be purged of unnecessary conjuncts.

3 Related Work

There is a large literature on causal discovery. Since this research concerns the causal regularity theory of causation, we restrict or review antecedents in the literature to work in that tradition. Although initiated by Hume and elaborated upon by J.S. Mill, the current *locus classicus* of the regularity theory is the article by (Mackie, 1964) [5], followed by his monograph (Mackie, 1984) [6]. Since then, Mackie's view has received a number of computational treatments. Baumgartner has provided philosophical justification of the regularity theory of causation and developed an algorithm restricted to configurations of Boolean values (Baumgartner, 2009 [2], 2009). Beirlan, Leuridan and Van De Putte support the idea computationally through a decidable subset of first order logic (Bierlan 2018) [3].

4 Method

4.1 A Brief Description

The purpose of our algorithm is to find cause-effect relationships implicit in datasets. The datasets used must consist of a table where each row is one observation, and each column represents an event. Each cell can signify that an event occurred, didn't occur (negated event) or that it's unknown whether or not it occurred. The algorithm starts with a chosen event and creates conjunctions (lists) of all events that also occur in the case that the chosen effect also occurred. Each conjunction that was generated is then tested to see whether it is necessary to the chosen effect occurring or not. A conjunction is necessary if it is not a superset of any of the other conjunctions. If it is a superset, then it is removed because there

* Department of Computer Science, 516 High Street. Emails: bidingt@wwu.edu, buzardh@wwu.edu, James.Hearne @wwu.edu, meinkea@wwu.edu, and tanners2@wwu.edu.

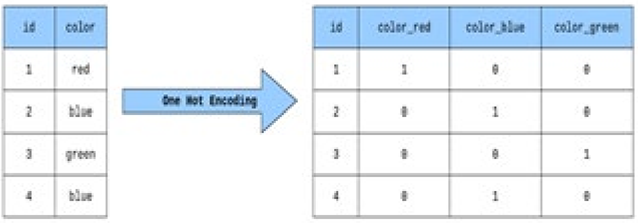
exists a simpler conjunction that better represents the data. These new conjunctions are then tested for sufficiency. Sufficiency is tested by ensuring that the chosen effect MUST occur if the given set of conjuncts also occurs. This is done by making sure that the set of conjuncts does not also appear in any of the rows where the chosen effect does not occur. Adding Figures and Tables.

4.2 In-Depth Account

4.2.1 Input Dataset: The algorithm assumes an ontology of individual objects and a collection of predicates which may or not be true of each. The algorithm also accommodates worlds in which predicate values are unknown for some objects. Assuming that the predicates will be relatively few and in order to accommodate an indefinite number of objects, it happens that columns correspond to predicates/attributes and rows denote the value of the predicates when applied to each object, including the possibility of unknown true values.

4.2.2 Formatting the Input Dataset: The dataset that is given as input is formatted according to how the algorithm expects the data to appear (only values of 1, 0, and -1 are accepted). This allows our algorithm to be versatile and make accurate computations for all datasets. The program prompts the user to label each column and choose whether to keep the column data as it is, remove the column from the dataset, or one-hot encode the data in the column. One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms using 1's and 0's. A visual representation of how to utilize one-hot encoding is shown in Diagram 1 below. The user is prompted to make these changes because our algorithm will not accept data that is not able to be represented with a 1, -1, or 0 and will ask the user to re-input the data if it finds a number not in this form.

4.2.3 Choose an Effect: The effect whose possible causes is in input to the process. The data is separated into two subsets: one object for which the effect is positive and one for objects in which the effect is negative, i.e., conditions in which it does not occur. The algorithm uses the former subset to generate potential MINUS-conditions, and the latter subset to check whether or not the generated potential MINUS-conditions are sufficient to prove the chosen effect.



id	color
1	red
2	blue
3	green
4	blue

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Diagram 1

4.2.4 Generate Possible Minus Conditions. To generate potential MINUS-conditions, the algorithm iterates through all rows in the data where the chosen effect obtains. For each of these rows, a set is created of all of the predicates in the row. All subsets are generated using this row data and each set that doesn't include the chosen effect and is not empty is added to the set of potential MINUS-conditions. It should be noted that this set of generated MINUS-conditions is a superset of the set that contains all sufficient conditions, and that the disjunction of the potential MINUS-conditions in this set is necessary for the chosen effect to occur (assuming there is more than one event type).

4.2.5 "Sufficient" and "Necessary" MINUS-condition check: Once the algorithm has generated the set of possible MINUS-conditions, each condition needs to be tested to check that it is sufficient and not a superset of a previously identified condition. By removing such supersets, the algorithm ensures that all parts of the proven MINUS-conditions are necessary. In order to prove that a given potential MINUS-condition is sufficient, the algorithm checks if the entire conjunction occurs in a row where the effect does not occur. Given that the conjunction being tested was previously known to occur when the chosen effect occurs, then if it also does not occur in a row where the chosen effect is negative, then it is proven to be sufficient for the given dataset.

4.2.6 Algorithm Output: Finally, all the identified MINUS-conditions are combined into a disjunctive normal form, presented as output. Each conjunction (MINUS-condition) is considered as a causal for the effect., one element of which would be what is normally identified as the cause of the event. It is important to note that there may be a case where it is impossible to have a necessary disjunction of conjunctions for a specific chosen effect (see Table 1 using R as the chosen effect). In Table 2, all data for rows 'a' and 'c' are equivalent except for the effect which occurs in one and doesn't occur in the other. Because of this, there is no MINUS-condition that includes row c. Therefore, no disjunction of MINUS-conditions exists which is necessary for R to occur in this dataset. However, if there is a disjunction of MINUS-conditions where the disjunction is necessary, each disjunct is sufficient, and each conjunct in each conjunction is necessary for the conjunction to be sufficient, then the algorithm will give this as output.

Table 1

	P	Q	R
a	1	-1	-1
b	1	1	1
c	1	-1	1

Table 2

	P	**Q**	**R**	**S**	**T**	**U**
a	1	-1	-1	0	1	-1
b	1	0	1	1	-1	1
c	-1	-1	0	-1	1	-1
d	1	-1	-1	-1	0	1
e	1	0	1	-1	-1	1
f	1	-1	-1	0	1	0
g	0	1	-1	-1	-1	0
h	1	-1	1	1	-1	1

5 Datasets

The datasets used for our algorithm contain columns which refer to events/attributes and rows that refer to observations. The value for an observation can be either 1 (event occurred), -1 (event did not occur), or 0 (unknown). Our algorithm is unique from other algorithms in the sense that it allows for the usage of a 0 or unknown in the dataset.

The following is an example of a dataset appropriate to our implementation that has been used in the development of this algorithm. In this case, considering the event 'P' we can see that all observations occurred except for the observation 'c' and there is no data provided for observation 'g.

A portion of the Cleveland Heart Disease dataset explored in greater depth below is shown in Table 3. This dataset is important to visualize and understand due to the fact that it can be used to gain actual knowledge and insight on the causation of a specific issue, unlike the dataset containing just letters above.

Table 3

Disease	Age	Sex	ind_typ_angina	
1	-1	63	1	1
2	1	67	1	0
3	1	67	1	0
4	-1	37	1	0
5	-1	41	0	0
6	-1	56	1	0

The columns of the dataset (this is only 4 of the 14 we used) indicate whether the patient was afflicted with heart disease, their age, their gender, and whether they experience typical angina (chest pain) or not. To understand one column of the data, for the sex columns, all 5 of the patients in this subset were male and the gender is unknown for one patient, represented as a '0'. We used this dataset to find conjunctions of conditions that are shown to cause heart disease. These results are discussed in the 'Results and Analysis' section below.

6 Datasets

6.1 Urinary Disease Dataset Description

6.1.1 Dataset Description: This data was designed to automate the decision making/diagnosis of the presumptive diagnosis of two diseases of the urinary system, "Acute Inflammation of Urinary Bladder" and "Acute Nephritis of Renal Pelvis". The dataset contains six attributes applicable to these two diseases, which have similar, but not identical symptoms. These attributes are as follows: fever present (at or above 38C), occurrence of nausea, lumbar pain, urine pushing (continuous need for urination), micturition pains (pain while urinating), and urethra discomfort (burning, itching, or swelling). The dataset also contains information on whether or not each patient associated with this data has one of the diseases, no diseases, or both diseases. Each instance (row) in the dataset represents a patient.

6.1.2 Dataset Description: There are certain symptoms known by experts to signify one or the other disease. One test of the utility of our algorithm for identifying causal regularities is to see whether it replicates expert knowledge. Dr. Czerniak, of the Polish Academy of Sciences, reports that Acute Inflammation of the urinary bladder is characterized by sudden occurrence of pains in the abdomen region, urination in the form of constant urine pushing, micturition pains, and sometimes lack of urine keeping. The excreted urine is turbid and sometimes bloody. The body experiences a temperature rise, however most often not above 38C. By contrast, Acute nephritis of the renal pelvis begins with a sudden fever that reaches and sometimes exceeds 40C. The fever is accompanied by shivers and one-or both-side lumbar pains. Not infrequently, there is nausea and vomiting and spread of pains in the whole abdomen. Again, symptoms of acute inflammation of the urinary bladder appear very often. Our dataset does not cover all of these characteristics/symptoms such as gender, shivers, and entire abdomen pain, however, given the data we do have, we should be able to match up applicable attributes. The algorithm will give us the combination of individual conditions that lead to each respectable disease and should pair well with the human expert findings.

6.1.3 What the Algorithm Identifies: When the algorithm is run using "Inflammation of urinary bladder" as the chosen effect, we receive the following proven conditions:

```
('~fever', 'urethra-discomfort')
('~fever', '~lumbar-pain')
('~fever', 'urine-pushing')
('~lumbar-pain', 'urethra-discomfort')
('~nausea', 'micturition-pains')
('~lumbar-pain', 'urine-pushing')
('nausea', 'urethra-discomfort')
('~fever', 'micturition-pains')
('micturition-pains', 'urethra-discomfort')
('nausea', 'urine-pushing')
('~lumbar-pain', 'micturition-pains')
('urine-pushing', '~urethra-discomfort')
('urine-pushing', 'micturition-pains')
```

When the algorithm is run using "Nephritis of renal pelvis origin" as the chosen effect, we receive the following proven conditions:

```
('nausea',)
('~urine-pushing', 'micturition-pains')
('~micturition-pains', 'urethra-discomfort')
('fever', 'urine-pushing')
('fever', 'lumbar-pain')
('lumbar-pain', 'urine-pushing')
```

('fever', 'urethra-discomfort')
('fever', 'micturition-pains')
('lumbar-pain', 'urethra-discomfort')
('lumbar-pain', 'micturition-pains')

These results are very promising and replicate closely the expertly deduced symptoms. As the output shows, some symptoms such as urethral discomfort are present more or less in both diseases, however when these symptoms happen in conjunction with lumbar pain or a fever, this always signifies nephritis, not inflammation. Likewise, if a patient has these symptoms and no lumbar pain or fever, they almost certainly have inflammation of the urinary bladder. There are also some conditions, mainly micturition pains, which seem to be only slightly more characteristic of inflammation over nephritis.

6.2 Heat Disease Dataset

6.2.1 Dataset Description: A promising dataset that both exhibits the accuracy of our algorithm and reveals important information regarding heart health, is the Cleveland Heart Disease dataset from the UCI repository. The Cleveland dataset is one of the most used datasets in Machine Learning and can be used to classify whether an individual is at risk for suffering from heart disease or not. The data was retrieved from 303 individuals and originally contained 76 columns, however, the shorter version of the dataset, which has been used for all the published experiments, contains only 14 columns. These columns are the 14 attributes that were found to have the biggest impact on classifying heart disease. These attributes chosen for the Machine Learning experiments are: age, sex (male or female), whether the patient was experiencing typical angina, atypical angina, or non-angina related chest pain (angina is chest pain caused by reduced blood flow to the heart), resting blood pressure, serum cholesterol level, fasting blood sugar (should be less than 120 mg/dl), heart disease flagged on ECG 1, heart disease flagged on ECG 2, patient's max heart rate, whether the patient experienced exercise induced angina, patient's peak exercise ST segment upward slope indicator, patient's down slope indicator of peak exercise ST segment, number of patient's major vessels colored by fluoroscopy, whether the patient has reversible thalassemia defect or fixed thalassemia defect (thalassemia is an inherited blood disorder).

To get results that show the proven conjunctions of attributes that cause heart disease, we ran the algorithm and gave 'disease' as input for the chosen effect. The algorithm checked 93,759 conditions in 2 minutes and gave 96 proven conditions as output.

6.2.2 What We Expect: Choosing 'disease' as the chosen effect, we would expect to see conjunctions of the following as causes of heart disease based on research by medical professionals:

Sex: Males are more likely to develop heart disease than women

Blood Sugar: A fasting blood sugar is expected to be between 80-100, and anything over 100 can be an indicator of multiple diseases or illnesses, one of those being heart disease

ECG Indicators: If an ECG indicates abnormal heart rates or an abnormal pattern once or especially twice, these are most likely signs of heart issues such as heart disease

Thai Fixed and Reversed Defect: This is a type of blood issue that can lead to organ failure as well as heart issues (one of the issues being heart disease). Fixed means that it is permanent and reversed means that the defect can be reversed, however, both of them can still cause organ damage and heart failure.

Fasting blood sugar: Over time, high blood sugar can damage blood vessels and nerves that control the heart which can cause heart attacks

Max heart rate: Numerous studies have shown that higher resting heart rate is associated with increased risk of cardiovascular events and death in men and women.

Upward slope indicator: ST-Elevation is very serious and can mean that one of the heart's major arteries is blocked. Even if the artery is not currently blocked, any abnormal ST-Elevation indicates risk of major artery blockage

Down slope indicator: ST-segment depression is associated with a 100% increase in the occurrence of three-vessel/left main diseases and to an increased risk of subsequent cardiac events

Number of patient's major vessels colored by fluoroscopy: Fluoroscopy is used to help the healthcare provider see the flow of blood through the coronary arteries to check for arterial blockages. The more blood vessels that show blockage, the higher the patient is at risk for heart disease and heart attacks

Serum Cholesterol: A high serum total cholesterol level has been proven to indicate a potential increased risk for heart disease.

Resting blood pressure, exercise induced angina and typical/atypical angina alone cannot accurately predict heart disease or risk of heart attack, which is why they are not listed above. However, either of these in conjunction with each other or other symptoms can be a proven indicator of a patient being at risk.

6.2.3 What the Algorithm Reveals: In order to best understand the results we received, we have analyzed a subset of the output to ensure that the algorithm output matches what we would expect to see based on expert research. Every conjunction below lines up with what expert researchers would say could be a cause of heart disease. Each condition that is not listed below was also analyzed for correctness.

('sex', 'bloodsugar_exc120', 'ind_for_ecg_2', 'ind_exerc_angina')

This conjunction is valid for being a direct cause of heart disease according to the research above. Sex can be a cause of heart disease because males have been shown to be more prone to heart disease and heart attacks than women. Having an indicator for heart disease show up on an ECG is also typically shown to be accurate and mean a patient is at risk for heart disease. Also, as I explained previously, an individual's blood sugar exceeding 120 and exercise induced angina cannot be a cause on its own, however in conjunction with each other and the other attributes, they can both be a proven cause.

('ind_atyp_angina', 'ind_exerc_angina', 'fixed_defect')

Using the information above, exercise induced angina and atypical angina cannot be a cause on its own but in this case both of those are paired together and also paired with the patient having thai fixed defect. Fixed thai defect is irreversible which means it usually eventually will lead to heart failure and in conjunction with also having chest pain occurring in multiple instances, this patient is rightfully flagged as being at risk for heart disease.

('ind_for_ecg_1', 'rev_defect')

Thai reversed defect on its own can be an indicator of heart disease even though it is reversible because it can cause organ damage to the heart. This defect paired with the patient being flagged for heart disease on their ECG puts the patients in an at risk category and the algorithm correctly identifies them as potentially suffering from heart disease.

6.3 Soybean Dataset

6.3.1 Dataset Description: One illuminating dataset is the one on soybean diseases from the 1980s by R.S. Michalski and R.L. Chilausky. The dataset concerns soybean disease diagnosis so we can use it to analyze the performance of our logic in discovering causal regularities. One of the important factors of this dataset is that there are many missing values/data points. Our algorithm is designed to work on datasets with missing data which makes this dataset a great example of this capability. Another important factor of this dataset is its limited number of datapoints. There are 307 data points (individual plants) and none of the 19 classes (different diseases) have more than 40 examples.

For this test, we only included things that we knew could be causes of anthracnose. The only predicates allowed were: temperature, precipitation, hail, and treatment type. This resulted in a dataset with 307 rows (plants) and 14 columns (event types)

6.3.2 What we Expect: We tested our algorithm by looking for things that prevent anthracnose, which is a fungal soybean disease. This test represents a scenario where a soybean farmer wants to prevent it in their crop, or just wants to know what can cause anthracnose. As you will see, these causes can be discovered by telling our algorithm to find the causes of anthracnose, and to find the causes of the lack of anthracnose. Since it is well known by soybean farmers that anthracnose is known to occur during warm, wet, and humid conditions, we expect our system to signal this fact.

6.3.3 What the Algorithm Actually Identifies: When the algorithm was run using anthracnose as the chosen effect, the algorithm returned no proven conditions. This means that there are no combinations of factors that guarantee anthracnose. This could be somewhat useful to a farmer or soybean researcher, but the more interesting findings are when the algorithm was run using the lack of anthracnose as the chosen effect. When the lack of anthracnose was the chosen effect, 154,295 conditions were tested, resulting in the following proven conditions:

```
[('temp_?',),
('precip_1',),
('temp_0',),
('precip_0',),
('~precip_2',),
('precip_?',),
('treatment_?',),
('temp_1', 'hail'),
('~temp_1', '~temp_2'),
('~temp_2', 'hail'),
('~temp_1', '~hail'),
('temp_2', '~hail'),
('~treatment_0', '~treatment_1', '~treatment_2')]
```

Since many of the predicates were implemented with one-hot encoding (See diagram 1), some of these conjunctions should be ignored (an underscore in the attribute name ex: temp_1, signifies that one-hot encoding was used). Some of these should be ignored because they are an artifact of one-hot encoding. For example, if we know that temperature is not high and not normal, we know that it must either be low or unknown, and this logical entailment in itself says nothing about the data set itself.

The principal takeaways from the output are that anthracnose never occurs under the following conditions: there is normal or less than normal precipitation, the temperature is less than normal, the temperature is normal and there is no hail, the temperature is greater than normal and there is hail or when the treatment is unknown. The treatment being unknown is odd since one might assume the data collectors would have that information, but it is unlikely to provide any useful information for this so it will be ignored for the analysis. One thing that should be considered when analyzing these results is that anthracnose is known to occur during warm, wet, and humid conditions. This only bolsters our results, since our algorithm showed that low temperature prevents anthracnose, and it only occurs when there is greater than normal precipitation. Since this is already known, we have shown the algorithm's ability to discover causal regularities.

7 Time Complexity Analysis

7.1 Algorithm Analysis

The algorithm is made up of 3 main parts: reading the dataset, generating possible MINUS conditions, and verifying or discarding all the possible MINUS conditions. Reading the dataset iterates over all predicates (P) and all data points (n). This gives a total time of O(nP).

When the algorithm generates the potential MINUS conditions, it does the following for every data point; prepares the dataset for analysis O(P), finds the set of all subsets from the data point's set of predicates O(2^P), and adds the members of this set to the set of all potential MINUS conditions O(2^P). Since there are n data points, generating MINUS conditions is O(n(P+$2^P + 2^P$)) which is equivalent to O(n2^P).

Establishing or rejecting each MINUS condition iterates over the set of all potential MINUS conditions, which can be as large as 2^P. For each of these potential MINUS conditions, we check that it is not a superset of any proven MINUS condition. Iterating over all proven MINUS conditions can be as large as O(2^P). Then each MINUS condition that survives this winnowing is checked against every datapoint. This takes O(nP^2) time for each potential condition it checks. The total time for this section is O($2^P(2^P + nP^2)$) which is equivalent to O(2^{2P}). However, since there are often a much smaller number of verified MINUS conditions in a real dataset, in practice the upper bound is often O(n$P^2 2^P$).

Combining all these sections results in a final time complexity of O(2^{2P}). However, in practice this is closer to O(n$P^2 2^P$).

7.2 Theoretical Minimum Time Complexity

This section calculates the maximum possible set of proven MINUS conditions. If it is assumed that an algorithm takes O(1) time to compute and output each MINUS condition, this yields a theoretical lower bound for an algorithm that generates MINUS conditions. However, such an algorithm would be unrealistic, so this will only be used as a way to analyze the algorithm described in this paper.

By way of proof, there are $\frac{P}{P/2}$ sets in the largest set of MINUS conditions. Since each one of these sets is no larger than P-1, if it takes O(1) to generate and output each part of each MINUS condition, the total time complexity would be O(P$\frac{P}{P/2}$). P$\frac{P}{P/2}$can be expanded to $P(\frac{P!}{(P/2)!(P/2)!})$. Using Stirling's approximation, P! is equivalent to $\sqrt{2\pi P}(\frac{P}{e})^P$ as P approaches infinity. However, it is more accurate to find the upper and lower bound of P $\frac{P}{P/2}$ using the upper and lower bound of Stirling's approximation. For the upper bound, the numerator will use the upper bound of Stirling's approximation $eP^{P+1/2}e^{-P}$, and the denominator will use the lower bound of Stirling's approximation

$$\sqrt{2\pi}(\frac{P}{2})^{\frac{P+1}{2}}e^{-\frac{P}{2}}$$

This will maximize the approximation of P$\frac{P}{P//2}$. Substituting these upper and lower bound approximations yields

$$P\frac{eP^{P+1/2}e^{-P}}{(\sqrt{2\pi}(\frac{P}{2})^{\frac{P+1}{2}}e^{-\frac{P}{2}}\ )^2}$$

Distributing the exponent in the denominator gives

$$P\frac{eP^{P+1/2}e^{-P}}{2\pi(\frac{P}{2})^{P+1}e^{-P}}$$

Canceling out e^{-P} , moving 2^{P+1} to the numerator and simplifying gives $\frac{e}{\pi}\sqrt{P}2^P$. Following the same process but with the lower bound of Stirling's approximation in the numerator and the upper bound in the denominator yields the lower bound for P$\frac{P}{P/2}$. This lower bound is

$$\frac{2\sqrt{2\pi}}{e^2}\sqrt{P}2^P$$

Since both the upper and lower bound are O($\sqrt{P}2^P$), the time complexity of this theoretical algorithm is O($\sqrt{P}2^P$).

As mentioned previously, this theoretical algorithm is unrealistic since it can find each MINUS condition in O(P) time, and it does not account for the number of data points. With this in mind, the algorithm described in this paper is mathematically O(2^{2P}), but in practice is closer to O(n$P^2 2^P$). Both algorithms take exponential time which shows that the algorithm described in this paper is within the same time complexity class as the theoretical minimum.

8 Conclusion

The identification of causal relationships in datasets can be very illuminating. There have been various approaches to the notion of causality, and through research and experimentation we have built upon these approaches to create a well-rounded algorithm for identifying causal relationships.

As shown in the results portion, we have proved that our algorithm successfully identifies singular and conjunctive conditions that serve as possible causes for a chosen event. The three datasets we have discussed exhibit the key features of why our algorithm is useful in finding these causal relations and also evidence of the accuracy of the algorithm that we have composed. The soybean dataset test illustrates the importance of finding causes for the inverse of an event and also illuminates the fact that our algorithm cannot distinguish between causes and symptoms of an effect. The algorithm finds correlations, but it is up to the user to only include things that would be causes and not symptoms, or the user would have to manually analyze the output and determine its likelihood of being a cause or symptom.

The causal regularities that the algorithm generates are reliably predictive because they are only produced if there is certainty, they will hold for a given data set. This predictive capability arouses a comparison to machine learning, arguably the most popular method for an algorithmic approach to prediction. Two of the biggest problems with machine learning are the need for lots of data, and the inability to see why a trained model makes a decision. Our algorithm doesn't suffer from either of these problems, but still creates causal regularities that can be used to make accurate predictions. More data is always useful for better prediction for both machine learning and our algorithm, but our algorithm requires a much smaller amount to get meaningful results. Even with only 20 examples of plants with anthracnose (a much too small amount for any standard machine learning algorithm) we were able to find useful information about the causes of anthracnose, and how to prevent it with certainty.

Another advantage of this algorithm is its 'white box' character; that is, it is straightforward to reconstruct the results obtained. This is useful for a dataset like the heart disease dataset because knowing the conditions and conjunctions of conditions that may cause heart disease can help individuals seek help sooner and also take care of their health in order to prevent themselves from being diagnosed with heart disease. Because we know that the causal regularities we generate will always hold true for the given dataset, then if we are confident that the dataset is fully representative of the problem, we will only generate ironclad predictions that will almost certainly always hold true.

Other methods of prediction can be effective for large datasets that don't require human understanding, but our algorithm can use small datasets to create precise predictions that are easily interpretable. Outside of prediction, our algorithm is also able to discover unknown properties in these datasets, generating new knowledge that other approaches could not gain.

9 Future Work

This implementation is open to sundry developments. First, to be a habitable system, a more congenial user interface is advisable. In addition, a redesign of the basic search for MINUS conditions along the lines of the *A Priori* algorithm for frequent item sets in market basket analysis would be possible. This *A Priori* approach would cut down on mathematical time complexity by a power of 2. However, in practice, this decrease in time complexity would be significantly lower. Beyond these matters of performance and cosmetic ease of use, its functionality might be augmented in several ways:

1. First, at present the algorithm detects MINUS conditions, i.e., conjunctions of conditions which are sufficient for the effect. It also assembles all such conditions such that, in the input data set, their disjunction is collectively necessary, i.e., one such conjunction must be present for the effect to occur. The algorithm does not explore the relative importance of each individual conjunct. Although most theories of causal regularity do not provide much guidance, we sense the possibility of isolating events of particular importance through abstraction; that is, two attributes may be instances of the same abstraction, and might be consolidated into a single event type. For example, if the data set has three attributes 'is colorless,' 'is green,' 'is blue,' and it turns out that the conjunction of 'is green' and 'is blue' figures in a MINUS condition, it might be abstracted to the simpler 'is colored'.
2. Metrics analogous support and confidence such as are found in itemset mining are appropriate and potentially useful. The algorithm will reliably organize the data into the conjunction of MINUS conditions, but they may apply only to a few data points and hence have little predictive force. Measure so confidence might be appropriate if we relax the stipulation that MINUS conditions are identified only if they have complete predictive power.
3. The existence of unknown values for certain predicates and objects, arouses the possibility of suggesting experimental designs. For example, discerning that a known positive or negative value for a given attribute would establish an additional causal hypothesis, could be useful in exploring the phenomenon summarized in the given data set. Also, discovering that the elimination of an attribute with many unknown values might lead to more definitive results could lead to greater insight into a data set.

References

[1] Michale Baumgartner, "Regularity Theories Reassessed," *Philosophical*, 36:327-354, 2008.

[2] Micheal Baumgartner, "Uncovering Deterministic Causal Structure: A Boolean Approach," *Synthese*, 170(1):71-96, 2009.

[3] Mathieu Beirlan, Bert Leuridan, and Frederik Van De Putts, "A Logic for the Discovery of Deterministic Causal Regularities," *Synthese*, 195:367-399, 2008.

[4] G. Graßhoff and M. May, "Causal Regularities, W. Spohn, M. Ledwig, and M. Esfield (Eds.) *Current Issues in Causatio*n, Paderborn: Mentis, pp. 85-214, 2001.

[5] J. L. Mackie, "Causes and Conditions," American Philosophical Quarterly, American Philosophical Quarterly, 2(4):245-264, Oct. 1964.

[6] J. L. Mackie, *The Cement of the Universe: A Study of Causation*, Oxford: Clarendon Press, 1984.

[7] J. S. Mill, *A System of Logic*, London: John W. Parker, 1843.

[8] Judea Perl, *Causality. Models, Reasoning and Inference*, Cambridge University Press, Cambridge, 2009.

Thomas Bidinger (photo not available) recently graduated with baccalaureate degrees in Computer Science from Western Washington University.

Hannah Buzard (photo not available) recently graduated with baccalaureate degrees in Computer Science from Western Washington University.

James Hearne (photo not available) is a Professor of Computer Science at Western Washington University, where his research is focused on the application of data mining and machine learning techniques to the interpretation of ancient historical records.

Amber Meinke (photo not available) recently graduated with baccalaureate degrees in Computer Science from Western Washington University.

Steven Tanner (photo not available) recently graduated with baccalaureate degrees in Computer Science from Western Washington University.