A Network-Based Deterministic Model for Causal Complexity

Henry Su
The University of Sydney
lvl2pillow@gmail.com

Niku Gorji
The University of Sydney
niku.gorji@sydney.edu.au

Simon Poon
The University of Sydney
simon.poon@sydney.edu.au

Abstract

Despite the widespread use of techniques and tools for causal analysis, existing methodologies still fall short as they largely regard causal variables as independent elements, thereby failing to appreciate the significance of the interactions of causal variables. The prospect of inferring causal relationships from weaker structural assumptions compels for further research in this area. This study explores the effects of the interactions of variables in the context of causal analysis, and introduces new advancements to this area of research. In this study, we introduce a new approach for the causal complexity with the goal of making the solution set closer to deterministic by taking into consideration the underlying patterns embedded within a dataset; in particular, the interactions of causal variables. Our model follows the configurational approach, and as such, is able to account for the three major phenomena of conjunctural causation, equifinality, and causal asymmetry.

1. Introduction

Studies of causal complexity play a pivotal role in many areas of research. In many ways, causal analyses often rely on correlational approaches. Such approaches fall short in accounting for the three major phenomena of conjunctural causation, equifinality, and causal asymmetry. However, through recent advances in causal analyses, a configurational (or set-theoretic) approach has brought about significant improvements to the performance of the causal analysis, especially for datasets with high dimensionality.

Nonetheless, as these advances were based on a methodology known as Boolean minimization, they are primarily focused on obtaining a minimal solution rather than high accuracy. This key aspect differentiates between causal complexity and causal analysis, which extends the functionalities to additionally extract useful configurational patterns from a dataset. As we shift from causal analysis and move to causality complexity, we also begin to draw inferences from our solutions and reason causation. Furthermore, previous approaches also fall short in producing consistent and deterministic solutions as they relied on strategies of random choice, i.e. the solution set produced is often non-deterministic.

Since the development of Qualitative Comparative Analysis (QCA) [1], the potential applications of causality analysis in many areas of research emerged as a viable option [2]. However, it was noted by several researchers that the underlying structure for QCA was vastly inefficient at dealing with large numbers of causal variables, thus severely restricting its application [2-5].

This performance bottleneck was resolved with the approach called BOOM [5]. BOOM opened the gateway to the study of causal complexities for many areas of research that dealt with large numbers of causal variables; however, also introduced new problems to the causality analysis. As these issues were ultimately resolved using non-deterministic strategies [5], their solutions were also non-deterministic.

Researchers argue that causation is not entirely attributed from the statistical configurations of causal variables[6]. They assert that weaker relationships, such as the interactions between causal variables, plays a role in determining the outcome of a particular configuration. Often, these relationships are trivialized due to the small number of causal variables, which makes it relatively safe to assume that each causal variable is an independent element. However, with increasing number of causal variables, the effect of each independent variable gradually becomes diluted, and conversely, the interaction of variables begins to play a significant role. Yet, these relationships are largely neglected in the existing methodologies, and as such, the solutions obtained through these methods are insufficient to accurately infer causation.

Regardless of the widespread use of causal analysis, existing methodologies largely regard causal variables as independent elements, thereby failing to appreciate the significance of the interactions between these elements. The prospect of inferring causal relationships from weaker structural assumptions compels for further research in this area.

URI: http://hdl.handle.net/10125/50094
ISBN: 978-0-9981331-1-9

This study explores the effects of the interactions of variables in the context of causality analysis.

Researchers utilizing causal analysis within their studies benefit in being able to perform analysis on high dimensional datasets, whilst still preserving a fast performance, and most importantly, achieving an accurate and reliable result.

Our main objective here is to introduce an approach for improving the accuracy of current causal analysis methodologies. By integrating concepts from sociology, such as centrality measures, our methodology introduces a new dimension for capturing the interactions of variables. The secondary objective is to develop a new efficient heuristic that would enable consistent and deterministic solutions for the causal complexity. Furthermore, the results generated could be easily interpreted and used for further reasoning by domain experts.

2. Previous Work

Although causal analysis lacks a formal definition, [1] in the fields of Social and Political Sciences described causal complexity as situations where "an outcome results from several different combinations of conditions". This was complemented as "a situation in which the effect of one variable or characteristic can depend on which others are present" [7]. At the core of all causality analysis is the notion of configurations. A configuration is a specific combination of elements that generates an outcome of interest [8]. From a configurational perspective, combinations of elements form various interconnected components which lead to a specific outcome.

This research uses set-theoretic approach of causal analysis. The analysis composed of two major steps: (1) Qualitative analysis and (2) Quantitative analysis. The qualitative analysis stage concerns cleaning and calibrating the dataset in to a Truth Table as shown in Table 1.

Table 1. Boolean representation of dataset.

v1	v2	v3	O
0	1	0	1
1	1	0	1
0	0	1	0
0	1	1	0
1	1	1	1

The quantitative step is the computational intensive task to extract all the possible configurations contributing to the outcome. This step relies heavily on the efficiencies of the Boolean minimization techniques. Hence, this is the focus of this study to improve the accuracy and efficiencies of the Boolean minimization process.

2.1. Terms and Definitions

Literal: Input variable v in the form (v or ~v).
Minterm: Product of terms i.e. configuration.
Implicant: Minterm that implies the desired outcome.
On-set: Set of minterms that lead to outcome of 1.
Off-set: Set of minterms that lead to outcome of 0.

2.2. Karnaugh Map

Karnaugh Map is one of the earliest techniques that provides a graphical method of minimization [9]. In this technique, for 'n' variables Boolean function, a map of n x n cells is constructed where each cell contains the outcome of corresponding configuration. The most notable advantage of this method is simplicity. This method makes the process of minimization significantly straight forward but becomes impractical for analysing more than 5 variables due to the visualization of the dimensions on the map.

2.3. Quine-McCluskey Algorithm

At the heart of all analyses for causal complexity is the study of configurations of variables [8]. One prominent methodology used to simplify logical expressions and extracts key configurations within a dataset is the Quine-McCluskey algorithm [11], [12]. It is described as "a partial solution to the problem of devising a mechanical method for simplifying truth functions"[13]. The algorithm is used for minimizing Boolean functions and is functionally identical to Karnaugh mapping, but uses a tabular form which allows for more efficient use in computer algorithms. Furthermore, the tabular form provides a deterministic way to check that the minimal form of a Boolean function had been reached. The algorithm involves two key steps:

(1) Finding all prime implicants of the Boolean function.

(2) Use those prime implicants to find the essential prime implicants of the function as well as other prime implicants that are necessary to cover the function. This was based on the solution to a set cover problem and formed the final simplified configuration of the truth table.

While the Quine-McCluskey algorithm has retained its status as a standard algorithm in Boolean minimization [2], its performance is far from efficient for the purposes of this study. The runtime complexity of the Quine-McCluskey algorithm is exponential and can be shown to approximate to $(3^n/n)$, where n is the number of variables in the input truth table.

Several studies have proposed modifications to optimize the existing Quine-McCluskey algorithm (e.g. Jan et al.[14]). The basis of these modifications were quite similar – to develop a modified method for generating prime implicants. By introducing new criterions when searching for prime implicants, the number of prime implicants generated in the first step was reduced. In this manner, the total number of comparisons of minterms in the Boolean truth table was reduced, thus improving the runtime of the algorithm overall. However, the optimizations presented from those papers only addressed a small portion of the inefficiency of the Quine-McCluskey algorithm.

2.4. BOOM Algorithm

BOOM, is a *heuristic* Boolean minimizer developed by Fiser and Hlavicka [5] and later improved in BOOM-II [10]. Similar to Quine-McCluskey, BOOM also includes the two basic stages of prime implicant gathering, and finding the solution to the covering problem. Where it differs is that the BOOM framework extended this by using a three-level bottom-up minimization strategy – these three stages are *coverage-directed search*, *implicant expansion*, and solving for the *covering problem*, respectively [5]. On top of this, BOOM takes advantage of the fact that most datasets were often large and sparse, and consisted of many don't cares. BOOM-II offered major improvements for functions with many output variables, but since our focus in on functions with many input variables, we take BOOM sufficient for the purpose of our study. To understand the working of BOOM, we consider the dataset in Table 2.

Table 2. Dataset showing on-set and off-set.

	v1	v2	v3	O
m0	0	1	0	1
m1	1	1	0	1
m2	0	0	1	0
m3	0	1	1	0
m4	1	1	1	1

Each row (record) is a minterm. The minterm giving the outcome of 1 is called 1-minterm and the one giving the outcome of 0 is called 0-minterm. The set of all the 1-minterms is called on-set and the set of all the 0-minterms is called off-set.

2.4.1. Coverage Directed Search (CDS)

The initial stage in the BOOM framework is the *coverage-directed search* (CDS). The algorithm used here searches for suitable literals, which are added in an iterative process to construct an implicant. The strategy is to start by picking the most frequent literal as it covers the largest proportion of the truth table. If the term being constructed does not intersect with the off-set, then it is classified as an implicant. Otherwise, a new literal is selected and added to the existing term, and the check for whether the new term intersected with the off-set continues. This process is repeated until the entire on-set is covered by implicants. The result of the CDS is a set of implicants, where each implicant is a covering of one or more minterms in a sum of product term. Collectively, the set of implicants wholly coveres the on-set.

The CDS could be executed through many iterations to increase the number of unique implicants. As the next stage is dependent on the quality of the implicant generation process, the more iterations that are run in the CDS, the better the final result [7]. However, the number of unique implicants generated declines as the number of iterations increases. In fact, the total number of implicants generated follows a logarithmic scale. As such, the nature of finding new implicants has a diminishing return and there exists a point where the trade-off between searching for more implicants and the runtime is no longer beneficial. Using Table 2 as the given data set, we have two implicants i.e. v1.v2 and ~v3. The output of this stage is shown in Table 3.

Table 3. Suppressing 1-minterms covered by v1.v2 & ~v3.

	v1	v2	v3	O
~~m0~~	~~0~~	~~1~~	~~0~~	~~1~~
~~m1~~	~~1~~	~~1~~	~~0~~	~~1~~
m2	0	0	1	0
m3	0	1	1	0
~~m4~~	~~1~~	~~1~~	~~1~~	~~1~~

2.4.2. Implicant Expansion (IE)

The *implicant expansion(IE)* is used to produce the set of prime implicants. A prime implicant is a subset of an implicant of minimal size in terms of number of literals, such that the removal of any literal from a prime implicant would result in a term that coveres one or more sets of data from the off-set. The algorithm in

this stage essentially tries to remove each literals from the implicant and if the new expression does not intersect with the off-set, then the literal removal is made permanent. There are 4 processing steps:

(3) Remove a literal and check if the new implicant intersects with off-set or not.

(4) If there is no intersection with off-set, make the removal permanent.

(5) Otherwise, put the literal back and select another literal for removal.

(6) Repeat the process till no removal is further possible.

This process reduces the size (length) of implicants, and is termed Implicant Expansion in the sense that the new implicant is probable to cover more 1-minterms (being shorter) thereby "expanding" the coverage of the implicant. Using the outputs generated from Covered Directed Search, we have two prime implicants i.e. v1 and ~v3 generated by IE.

2.4.3. Covering Problem solution (CPS)

The final stage is a heuristic solution to the covering problem. Fiser & Hlavicka [7] argued that an exact solution to the covering problem is time consuming and that a heuristic approach is the only viable method. This heuristic is called Least Covered, Most Covering (LCMC), whereby prime implicants covering minterms covered by the least number of other prime implicants, are preferred. In the event of a tie, the prime implicant which covers the most number of minterms that are not yet covered is chosen. The aim is to produce the minimal set of prime implicants called essential prime implicants. For CPS:

(1) Select the prime implicants that cover such 1-minterms which are covered by least number of other implicants. This heuristic is also called LCMC (Least Covered, Most Covering) heuristic.

(2) If there are more than one such (prime) implicants, implicants covering the highest number of yet uncovered 1-minterms are selected.

Table 4. Suppressing 1-minterms covered by ~v3.

	v1	v2	v3	O
m0	0	1	0	1
m1	1	1	0	1
m2	0	0	1	0
m3	0	1	1	0
m4	1	1	1	1

Continuing with the results generated by Implicant Expansion in the previous step, we select the implicant covering the minterm covered by least number of other implicants. Both v1 and ~v3 cover the 1-minterms (m4 and m0 respectively) and covered by least number of other implicants. Since there is a tie, the implicant covering the highest number of yet uncovered 1-minterms is selected. We can see that ~v3 covers 2 minterms (m0 and m1) and v1 also covers 2 minterms (m1 and m4). Since, there is another tie, next choice is random (say ~v3). Now, the 1-minterms covered by ~v3 are temporarily removed from the on-set. Since, ~v3 covers m0 and m1, we suppress those minterms (as shown in Table 4). Thus, we consider only the remaining 1-minterms i.e. m4.

Table 5. Suppressing 1-minterms covered by ~v3 and v1.

	v1	v2	v3	O
m0	0	1	0	1
m1	1	1	0	1
m2	0	0	1	0
m3	0	1	1	0
m4	1	1	1	1

We apply LCMC in the reduced on-set. v1 covers the minterm covered by least number of other implicants. Thus v1 becomes another essential prime implicant. Now, we temporarily remove the 1-minterm covered by v1, i.e. m4. The final essential prime implicants are: ~v3 and v1 generated by this final CPS step.

3. A Network-Based Deterministic Model

This proposed model builds upon the BOOM framework, and as such, it structure shares many similarities. It inherits the fundamental advantages of a configurational approach over its correlational counterpart. Additionally, this model introduces new advances in regards to the social aspects (i.e. taking advantage of the relatedness of the causal factors), which in turn leads to a more complete and accurate solution. And on top of this, it adapts the original methodology to a heuristic-guided exhaustive process, which guarantees a deterministic solution.

3.1. Coverage Directed Search

The CDS stage is necessary to generate a set of implicants that collectively cover the entire on-set without intersecting with the off-set. Each implicant consists of multiple literals (input variable in true or false form i.e. v or ~v), which are selected based on a

heuristic that combines the idea of literal frequency (as in BOOM) and social value of the literal in the dataset. Once the on-set is covered, additional implicants can be found to improve the quality of the overall analysis [5]. However, as finding the complete set of implicants has an immense overhead, only a subset of good implicants is searched for.

Notice that in BOOM, the method ultimately retires to a random choice should there be multiple candidate literals. In fact, BOOM relies on this randomness to enable each iteration of the search to produce a slightly different set of implicants. If there were no randomness, each iteration would produce the exact same set of implicants, thereby making any additional iterations redundant. This approach trivialised the importance of the social aspect, inevitably failing to capture much of interactive relationships in a dataset.

To address these issues, our method modifies the iterative search process to a heuristic-guided exhaustive process. Through this, a strategy that relied on random choice was entirely avoided. Although debatable that an exhaustive approach could render an immense performance overhead, we justified our choice that with a "good enough" heuristic, the search would still be as efficient. From our preliminary results, we experienced minimal performance drop.

As different datasets will contain different patterns, the interactions between causal variables will also vary. Thus, it is preferable to have a mechanism for controlling the behaviour of the heuristic to more accurately reflect the nature of the dataset. We introduce a new concept called *interactivity*, which adjusts the heuristic such that it can be more influenced by either the literal frequency or the social scores. For highly interactive datasets we can shift the bias in favour of the social scores, whereas for independent datasets, the bias would favour the literal frequencies. The following pseudocode outlines the method:

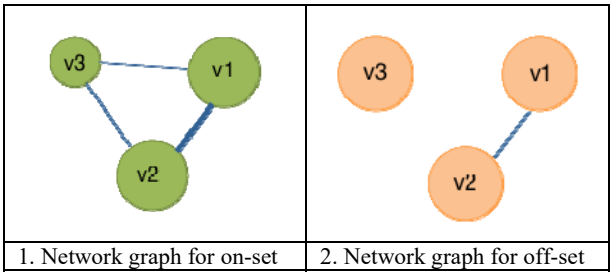
```
function cd_search(ON, OFF) {
      I = ∅
      // all permutations of the uncovered onset
      U = Queue()
      U.push(copy(ON))
      while |U| > 0
            // temporary set of implicants
            I′ = ∅
            U′ = U.pop()
            // initial term is empty
            construct_term(I′, ∅, copy(U′), OFF)
            I = I ∪ I′
            ∀ i ∈ I′
                  // update the current onset covering
                  U.push(U′ ∪ i)
      return I
}

function construct_term(I′, t, U′, off) {
      // if the current term intersects with the offset
      if t ∩ OFF ≠ ∅
            L = best_literal(t, ON)
            ∀ l ∈ L
                  // recurse until the term doesn't intersects with the offset
                  construct_term(I′, t′, U′ ∩ t·l, OFF)
      else
            I′ = I′ ∪ t
}
```

3.1.1. Social Score Heuristic

To determine the social value of each variable, a network graph G = (V, E) is formed where each input variable (causal factor) is considered as a node $v_i \in V$ and if two variables both are present in a row (minterm) leading to the outcome is allocated an edge on the graph $e_i \in E$. Network graphs for on-set and off-set for of dataset shown earlier in Table 2 are presented below.

Figure 1 shows the network diagram for on-set and off-set of the example dataset. Each variable is now expressed as a node in the network. In Figure 1.1, for each 1-minterm, an edge is drawn between two variables if both are 1 in the minterm. The bold link between v1 and v2 shows that they participate more together to produce the outcome of 1. Also, the larger size of v1 and v2 denote that they have higher degree in the network.

In Figure 1.2, Similar to the network diagram for on-set, variables are expressed as nodes in the network. For each 0-minterm, an edge is drawn between two if both variables are 0 in the minterm. As you can see, v3 doesn't have any link attached to it. It means that v3 doesn't interact with other variables to produce the absence of the outcome while the link between v1 and v2 denotes that the absence of v1 and v2 together in the dataset may lead to the absence of the outcome.



1. Network graph for on-set | 2. Network graph for off-set

Figure 1. Network graph

The social value score of each variable is calculated taking into account the *degree* centrality and *betweenness* centrality. Degree is the measure of number of edges connected to a node and Betweenness is the total number of shortest paths from all the existing vertices to all of the other vertices that pass through that particular node. Social value score of a variable is calculated as:

$$SS(v)=C_D(v_{on})+C_B(v_{on})- C_D(v_{off})- C_B(v_{off})$$

Where,

$C_D(v_{on})$: degree centrality of the vertex (literal) in on-set.

$C_B(v_{on})$: betweenness centrality of the vertex in on-set.

$C_D(v_{off})$: degree centrality of the vertex in off-set.

$C_B(v_{off})$: betweenness centrality of the vertex in off-set.

Then, the overall heuristic score of a literal $O(x_i)$ is calculated as:

$$O(x_i)=\theta\times SS(x_i)+(1-\theta)\times LF(x_i)$$

Where, θ denotes the interactivity, such that $0<\theta<1$, and SS is a function for the social score, and LF is a function for the literal frequency.

Now, we reveal a common special case which occurs when literals have the same overall score. Because the social score applies to the entire variable (both the original literal and its complimented form), it is not uncommon for both literals to have the same literal frequency, and subsequently, the same overall score. In these special circumstances, we prefer to select the original literal over is complimented form. We reasoned that the knowledge of the presence of a variable would be more useful than the absence of one.

3.2. Implicant Expansion

The next stage of the analysis is the implicant expansion, whereby each implicant from the CDS is reduced to its minimal form, and in doing so, creates a prime implicant. The process involves trying to remove each literal from the implicant, and checking if the resulting term intersects with the off-set. If the resulting term does not intersect, then the literal removal is permanent, else the removal is undone.

In BOOM, the literal selected for removal was by random choice. Consequently, the prime implicants formed were not deterministic, especially for long implicants. We incorporate a heuristic, which is essentially a greedy approach to select the worst literal for removal from our implicants. By "peeking" into the resulting term after the removal of a literal, we are able to assess its closeness with the off-set. Preferably, we'd want to remove a literal such that the remaining term is least similar to the off-set. Conceptually, we'd like to find a pattern which is least like the off-set, whilst still covers the on-set. To assess the closeness with the off-set, we use the Manhattan distance, where for 2 vectors u and v:

$$d_M = \sum|u_i - v_i|$$

We compare the Manhattan distance of the resulting term with each minterm from the off-set and select the minimum distance. If this minimum distance is equal to 0, then the resulting term would intersect with the off-set, and as such, the removal cannot be made. If a tie exists, the social score heuristic is used to break the tie, whereby the literal with the lowest social score is preferred for removal. And should another tie exist, like the CDS, an exhaustive approach is applied, whereby all equally worse candidates are tried for removal. This, in some instances may generate multiple prime implicants from a single implicant.

```
function implicant_expansion(I, OFF) {
	// set of Prime Implicants
	PI = ∅
	∀ i ∈ I
		ie_helper(PI, i)
	return PI
}

function ie_helper(PI, i) {
	// all literal candidates for removal
	L = worst_literal(i, OFF)
	// stop when no more removals are possible
	if L ≠ ∅
		∀ l ∈ L
			i = i.remove(l)
			ie_helper(PI, i)
	else
		PI = PI ∪ i
}
```

Here we see that we rank the worst literal based on 2 heuristics: firstly the Manhattan distance, and then the social score:

```
global S = social_score(ON, OFF)

function worst_literal(i, OFF) {
	// heuristics used to rank literals for removal
	H = {l: (min_difference(i.remove(l), OFF), S[l]) ∀ l ∈ i}
	// best heuristic value; sort by min_difference, social_score
	h = H.values().sort(key = x: -x[0], x[1])[0]
	return l ∀ l ∈ i if H[l] = h
}

function min_difference(i', off) {
	return min(manhattan_distance(i', m) ∀ m ∈ OFF)
}
```

3.3. Covering Problem Solution

The final stage involves solving for the covering problem, where a minimal set of prime implicants that wholly covers the on-set is found. Traditionally, this problem would be an instance of an NP-complete problem, known as the Unate covering problem. An exact solution to the covering problem would be time consuming and that a heuristic approach was the only viable method [5].

In BOOM, a heuristic called the Least Covered, Most Covered (LCMC) heuristic was used. Under this heuristic selected, prime implicants covering the most minterms that were covered by the least number of other prime implicants were favoured. If there were multiple such prime implicants, then the one which covered the most number of minterms not yet covered was preferred. Should another tie exist, then the shortest prime implicant was selected, i.e. the prime implicant with the least number of literals. Both

BOOM and our model utilise an adaptation of this heuristic.

In addition to this, we introduce a new Weighted Literal, Weighted Output heuristic (WLWO). Unlike the LCMC heuristic, the WLWO heuristic was designed for the sole purpose of logic minimisation and took into account of the relationship between implicants and minterms[15]. Further details about this heuristic are explained in the Weighted Literal, Weighted Output section below. In this research, we also adopt the WLWO heuristic.

Finally, we also introduce a new Weighted Social Score (WSS) heuristic, which as the name implies, is the derived from the social score of each literal from the prime implicants. With these heuristics, in most cases we were able to distinctly rank each prime implicant (in fact, in all of our tests, no ties ever occurred). Should a tie ever occur though, we reason that because the prime implicants were so similar, then selecting either would have been acceptable. As such, the final tie breaker is the order in which the prime implicant appeared. The following pseudocode outlines the implementation for the covering problem. It utilises a cover_matrix, which is a matrix where each row represents a minterm from the on-set, and each column represents a prime implicant. Then the values are either `0` – the prime implicant does not intersect with the minterm, or `1` – the prime implicant intersects with the minterm.

```
function unate_cover(PI, ON) {
	EPI = ∅
	C = cover_matrix(PI, ON)
	while C ≠ ∅
		// single best prime implicant candidate
		pi = best_pi(PI, ON, CM)
		EPI = EPI ∪ pi
		// update the cover matrix
		C = C.remove_pi(pi)
	return EPI
}

function cover_matrix(PI, ON) {
	// a cover matrix summarises the minterms that each prime implicant covers
	C = [[1 if m ∩ pi ≠ ∅ else 0 ∀ pi ∈ PI] ∀ m ∈ ON]
	return C
}
```

3.3.1. Weighted Literal, Weighted Output

The WLWO heuristic defines two key weights:

Weight of Literals (LW): Defined as the number of prime implicants which contained such a literal.

Weight of Outputs (IC): Defined as the number of implicants in the on-set or don't-care-set for each output. In our case with only a single output function, this is simply the cardinality of the on-set.

In addition, two weight functions are defined as follows:

Weighted Literal Count (WL_i): This is the weighted sum of the literals that are present in the implicant.

$$WL_i = \sum_{(x \in X_i)} (LW_x)$$

where X_i is the set of literals in implicant i.

Weighted Output Count: The summation of weights of outputs that contain implicant i.

$$WO_i = \sum_{(y \in Y_i)} (IC_y)$$

Where Y_i is the set of outputs that contain implicant i.

The WLWO heuristic is then defined as:

$$(WLWO)_i = (WL)_i \times (WO)_i$$

3.3.2. Weighted Social Score

We introduce another heuristic for the covering problem, thereby reducing the possibility of a tie even further. The Weighted Social Score (WSS) of a prime implicant is the sum of the Social Scores of each literal of that prime implicant, divided by the number of literals. We reasoned that this captured the "social" influence of the prime implicant as a whole, which corresponds to the central themes of our model.

The Weighted Social Score is defined as:

$$(WSS)_i = \left(\sum_{(x \in X_i)} (SS(x)) \right) / (|X_i|)$$

Where X_i the set of literals in implicant i, and SS is the social score. The pseudocode shown below summarises how each heuristic is calculated, and how they are used to rank each prime implicant:

```
global S = social_score(ON, OFF)

function best_pi(PI, ON, C) {
	// weighted literal, weighted output for each prime implicant
	WLWO = wlwo(PI, ON)
	// weighted social score for each prime implicant
	WSS = wss(PI)
	// heuristic used to rank prime implicants
	H = {pi: (sum(C[pi]), |pi|, WLWO[pi], WSS[pi]) ∀ pi ∈ PI}
	// best heuristic value; sort by cover, length, wlwo, wss
	h = H.values().sort(key = x: -x[0], x[1], -x[2], -x[3])[0]
	// return first prime implicant with best heuristic value
	return pi if H[pi] = h
}

function wlwo(PI, ON) {
	// all literals
	L = [l ∀ l ∈ pi ∀ pi ∈ PI]
	// literal weights
	lw = {l: L.count(l) ∀ l ∈ L}
	return {pi: sum(lw[l] ∀ l ∈ pi)*|ON| ∀ pi ∈ PI}
}

function wss(PI, ON, C) {
	// mean social score of literals in prime implicant
	return {pi: mean(S[l] ∀ l ∈ pi) ∀ pi ∈ PI}
}
```

4. Data

We considered two synthetically generated datasets of 100 variables (column) by 1000 configurations (rows). One dataset for which the configurations and outcome was randomly assigned assuming no relation exists among any of the variables and outcome. This dataset was regarded as independent dataset. For another dataset all the possible interactions of variables was considered and outcome was assigned based on the occurrence of interactive combinations.

(1) Independent dataset - where the input variables (casual factors) of the outcome (effect) are independent of each other and presence/absence of one factor doesn't affect the contribution of another variable in the causal relation; and;

(2) Interactive dataset – where the factors are highly interactive and contribution of a factor to the outcome is affected by the presence or absence of another variable.

5. Analysis

The two datasets were feed into BOOM and our model to test for accuracy, 30 percent (300 out of 1000 records) were randomly retained as holdout for comparing between the BOOM and our model. The implicants generated from the 700 records were tested using the holdout set. Accuracy, is calculated as:

$$\text{Accuracy} = (N_c / N_o) * 100\ \%$$

Where, N_c is the no. of 1-minterms covered by the implicants and N_o is the no. of 1-minterms in the original on-set. Results obtained are listed in the Table 6 and Table 7:

Table 6. Comparison for Independent Dataset.

Independent Data	BOOM	Our model
Run time (sec)	1.73	7.64
Average length of implicants	7	6.5
No. of implicants	58	81
Accuracy (%)	84.62	86.54

Table 7. Comparison for Interactive Dataset.

Interactive Data	BOOM	Our model
Run time (sec)	1.89	6.94
Average length of implicants	7	6
No. of implicants	61	79
Accuracy (%)	79.85	89.54

Experimental results show that though our model has a longer runtime compared to BOOM. It is perhaps due to the additional computational overhead for ensuring deterministic solutions are achieved. It has to be noted that our model is able to achieve a better accuracy in comparison with BOOM, especially for the dataset that was interactive.

6. Discussion

To evaluate our hypothesis, we must first know of the patterns within a dataset. However, as this is not feasible with real datasets, even with expert knowledge, the next best approach is to generate synthetic datasets with predefined patterns.

Essentially, we defined a score for each distinct pattern within the dataset. Conceptually, a pattern consisting of a single causal variable represents its independent value, whereas a pattern consisting of a combination of causal variables represents its interactive value – that is, the interactions between variables. Each pattern was associated with a relative score; positive scores marked a pattern for achieving a desired outcome, and the converse was also true for negative scores.

There were several qualities which we assessed: minimal solution, coverage, and accuracy, reliability (deterministic, consistency), performance. The minimal solution refers to the number of essential prime implicants generated by the analysis. Preferably, we'd like to have the most concise and minimal solution that summarises the core variables and patterns of the dataset. The coverage is the proportion of minterms covered by each PEPI (Positive Essential Prime Implicants). The PEPI consists of only the original literals from each EPI. The complimented form of each literal are therefore discarded. We measured the coverage for both the on-set and off-set. Preferably, we'd like a high coverage over the on-set, and a low coverage over the off-set. The coverage abstractly measures how significant the solution is. Lastly, the accuracy abstractly represents how well each PEPI captures the underlying patterns from the dataset – that is, our predefined patterns. We used the overlap to measure accuracy, which was explained in the previous section.

6.1. Strengths

6.1.1. Advantages over Correlational Approaches

In the Literature, we visited the issues concerning the phenomena of conjunctural causation, causal asymmetry, and equifinality[1], [16]. Traditional correlational approaches were limited as they were unable to interpret interactions beyond two-way effects

[16]. However, our model was built upon the notions of configurations, whereby multi-way interactions were the norm. As such, our model is able to account for the phenomenon of conjunctural causation.

Equifinality refers to situation where "a system can reach the same final state, from different initial conditions and by a variety of different paths" [17]. A methodology such as regression assumes *unifinality*, whereby a single optimal solution exists. In contrast, we have shown that our framework produces multiple configurations, where each can be equally effective. As such, we can conclude that our model is also able to account for the phenomenon of equifinality.

Correlational approaches assume a symmetry in the outcome, that is, if the presence of a condition leads to a particular outcome, then the absence of that condition must also lead to the inverse of said outcome [18]. However, this is not always true as causation in datasets are asymmetric in nature. This is reinforced by [19], who explains that "variables found to be causally related in one configuration may be unrelated or even inversely related in another", thus the phenomenon of causal asymmetry. As our model follows a configurational approach, it does not assume linearity, and therefore enables for asymmetric formulation.

6.1.2. Performance on High Dimensional Datasets

One of the key advantages of this new approach is its significant gains in performance on high dimensionality datasets compared to previous methodologies in causality analysis, such as the QCA framework. The runtime of QCA is exponential to the number of causal variables within the dataset. In comparison, our framework scales at a much lower rate. Through our results and evaluation, we determined that the rate at which the time scaled was sustainable with respect to the number of causal variables in the dataset, thus enabling large-scale causality analysis.

6.1.3. Identifying Core Components and Interactions

Our research accounts for the causal cores which can be defined as the resources that are most critical to the success of a particular event. One of the major advantages of set-theoretical approaches to causality analysis is the ability to easily identify core components of a dataset pertaining to the success of the relevant event.

In addition to this, our model extends the ability of identifying core causal variables by also capturing the underlying patterns within a dataset, including the interactions between causal variables. With a dedicated social component, our model is more suited to capture even the trivial relationships that exists within a dataset. This is demonstrated in our results, which revealed that our framework was able to achieve better accuracy in capturing the underlying patterns of a dataset, especially for interactive dataset, where the relationships between causal variables were the dominant factor for determining an outcome.

These findings mark a significant progress for this study, and places our model and the framework as a viable and improved alternative to causality reasoning than compared to existing methodologies.

6.2. limitations

Limited diversity refers to the phenomenon whereby particular configurations are not present throughout the dataset and may consequently impact upon the causality analysis. It has been argued that it "places severe constraints on possibilities for testing causal arguments. Because of limited diversity, statements about causation are necessarily restricted to the combinations of causally relevant conditions that actually exist"[1]. The existence of limited diversity in datasets, especially high dimensional datasets, can obscure key patterns from the causality analysis. As we witnessed, in sparse datasets such as the TCM (Traditional Chinese Medicine) dataset, the majority of configurations produced by the causality analysis largely consists of the complimented form of a literal, i.e. the absence of certain causal variable. An earlier version of this technique gas been successfully applied to study the effectiveness of combinations of herbs (as configurations) in TCM prescriptions in patient data records [22]. While in our interpretations, we have largely discarded these terms and only considered causal variables which are present, these configurations still entail meaning about the analysis. In order to produce results which are more meaningful and accurate, causal variables whose genuine absence forms as part of a configuration must be able to be distilled from the other causal variables.

The limitations of limited diversity pertains to all causality studies, including ours and warrants for further research.

7. Conclusion

We have explored the history of causality studies, stopping to examine each new generation of causality analysis; from Quine-McCluskey [11], [13] and QCA [1], to BOOM [5] and we now arrive at our new

approach. We've critically evaluated previous methodologies and improved upon their limitations.

Like many of our predecessors, our model follows a set-theoretic configurational approach, and as such, is able to account for the three major phenomena of conjunctural causation, equifinality and causal asymmetry, which correlational approaches lacked [1], [16]. Inspired by the works of [20], [21], we extended previous methodologies to comprehensively incorporate the social aspects of the causality analysis, thereby introducing a new class of social-enabled causality reasoning. We postulate that our model is more capable at capturing the patterns embedded within a dataset, which includes the causal variables' independent value as well as the interactions between causal variables, and this is reinforced through our results. In addition, through the integration of the social aspects, our model opened new avenues to achieving a deterministic solution. Whereas previous methodologies relied on an iterative process that consisted of random choice, our adaptation is a heuristic-guided exhaustive process, which produces deterministic solutions. It is important to note that the aim is to generated set-theoretic solutions to be interpretable by users, as each configuration in the final can then be verified by domain experts.

Through our efforts, researchers from many areas of research are able to benefit in being able to perform causality reasoning on high dimensionality datasets and achieve accurate and reliable results, whilst still preserving a fast performance. Furthermore, our model was designed to be modular, and as such, allows for future improvements and can be easily tailored to specific domains.

8. Acknowledgement

We would like to acknowledge the research contributions made by Li Bin from Nanjing Technical University (email address libin@njtech.edu.cn in implementing the algorithm into a software prototype during his visit to the University of Sydney between January 2016 and January 2017.

9. References

[1] C. C. Ragin, "The Comparative Method. Moving Beyond Qualitative and Quantitative Strategies," Berkeley: University of California Press, 1987.

[2] A. Thiem and A. Duşa, "QCA: A package for qualitative comparative analysis," R Journal, vol. 5, pp. 87-97, 2013.

[3] S. J. Hong, R. G. Cain, and D. L. Ostapko, "Mini: A Heurstic Approach For Logic Minimisation," IBM Journal of Research and Development, vol. 18, pp. 443-458, 1974.

[4] P. C. McGeer, J. V. Sanghavi, R. K. Brayton, and A. L. Sangiovanni-Vicentelli, "ESPRESSO-SIGNATURE: a new exact minimizer for logic functions," IEEE Transactions on Very Large Scale Integration Systems, pp. 432-440, 1993.

[5] P. Fišer and J. Hlavička, "BOOM - A heuristic Boolean minimizer," Computing and Informatics, pp. 19-51, 2003.

[6] J. Pearl, Causality: models, reasoning, and inference. Cambridge: Cambridge University Press, 2000.

[7] R. Jervis, System effects: complexity in political and social life. Princeton, N.J: Princeton University Press, 1997.

[8] B. Rihoux, et.al., "From Niche to Mainstream Method? A Comprehensive Mapping of QCA Applications in Journal Articles from 1984 to 2011," Political Research Quarterly, pp. 175-184, 2013.

[9] M. Karnaugh, "The Map Method for Synthesis of Combinational Logic Circuits," Transactions of the American Institute of Electrical Engineers vol. 72, pp. 593–599, 1953.

[10] Fiser, Petr, and Hana Kubátová. "Flexible two-level Boolean minimizer BOOM-II and its applications." Digital System Design: Architectures, Methods and Tools, 2006. DSD 2006. 9th EUROMICRO Conference on. IEEE, 2006

[11] W. V. Quine, "The Problem of Simplifying Truth Functions," The American Mathematical Monthly, 1952.

[12] W. V. Quine, "A Way to Simplify Truth Functions," The American Mathematical Monthly, v62, pp. 627-631, 1955.

[13] T. K. Jain, D. S. Kushwaha, and A. K. Misra, "Effect of Quine-McCluskey simplification on Boolean space complexity," in Fourth International Conference on Autonomic and Autonomous Systems, 2008, pp. 165 - 168.

[14] A. Kagliwal and S. Balachandran, "Set-Cover Heuristics for Two-Level Logic Minimization," 2012, pp. 197-202.

[15] P. C. Fiss, "A Set-Theoretic Approach to Organizational Configurations," The Academy of Management Review, vol. 32, pp. 1180-1198, 2007.

[16] C. C. Ragin, Redesigning social inquiry: fuzzy sets and beyond. Bristol; Chicago, Ill: Univ. of Chicago Press, 2008.

[17] A. D. Meyer, A. S. Tsui, and C. R. Hinings, "Configurational Approaches to Organizational Analysis," The Academy of Management Journal, vol. 36, 1993.

[18] S. K. Poon, K. Fan, J. Poon, C. Loy, K. Chan, X. Zhou, et al., "Analysis of herbal formulation in TCM: Infertility as a case study," in International Conference on Bioinformatics and Biomedicine Workshops, 2011, pp. 868-872.

[19] X. Zhou, J. Poon, P. Kwan, R. Zhang, Y. Wang, S. Poon, et al., "Novel two-stage analytic approach in extraction of strong herb-herb interactions in TCM clinical treatment of insomnia," 2010, pp. 258-267.

[20] S.K. Poon, A, Su, L. Chau, et al., "Causal Complexities of TCM Prescriptions; Understanding the Underlying Mechanisms of Herbal Formulation", Data Analytics for Traditional Chinese Medicine Research (Poon, J. and Poon, S.K. Ed.), Springer Publishing, 2014, pp.17-38.