Causal Networks and Their Decomposition Theories

Zhi-Qiang Liu

Abstract

Causal networks (CNs) have been used to construct inference systems for diagnostics and decision making. More recently, Bayesian causal networks (BCNs) and fuzzy causal networks (FCNs) have gained considerable attention and offer an alternative framework for representing structured human knowledge and are used in causal inference in many real-world applications. However, for large systems, it is difficult to analyze and design causal networks. This paper presents two new approaches to partitioning fuzzy causal networks: causal modules and quotient space.

Keywords: Causal Networks; Causal Modules; Quotient Space; Directed Graphs; Inference; Decision Making; Dynamic Systems

1. Introduction

In most dynamic systems, we must perform data analysis based on which we can come out with necessary strategies to achieve the desire effects. To make a decision, we must know the nature of the problem to be solved. For instance, in order to effectively solve the ozone layer depletion problem, we must identify their causes that help us to make laws and regulations to curb emissions that are harmful to the ozone layer. For such problems, two different processes are involved, namely, classification which produces labeled data given the features [1]; and causal prediction which is concerned with the effect of, the changes in some features to some other features in the system. For example, to classify vehicles, we may use color, size, and shape as its features, based on which we can label a particular object in the image as a track or a car. In many applications, we may use classification to gain new knowledge. For example, in searching for anti-AIDS vaccine, researchers use data from a certain population that has exposed to HIV but not developed AIDS to find a procedure to develop an effective anti-AIDS vaccine. In such problems, we are more concerned with a correct labeling process. As a major topic in machine learning, many techniques have been developed, e.g., neural networks, decision trees, various learning paradigms, such as competitive learning, supervised learning, reinforcement learning, and so on.

In causal prediction, we are concerned with the change of features and its effects, e.g., the impact of the increased piracy activities in the Internet on the entertainment industry; and the effect of a new Euro-currency to the global economy. To know the effects of the changes, we must have some mechanisms that are able to discover the cause and effect relations from the data set and handle uncertainties present in the data. During the 80s two major causal networks were developed, Bayesian causal networks [8] and fuzzy cognitive maps [6], and have since found many applications.

This paper is organized as follows: in Section 2, I briefly discuss Bayesian causal networks (BCN), especially, the D-separation theory. Section 3 presents fuzzy causal networks (FCN) and the theory of causal modules and algorithm for generating inference patterns. Section 4 introduces the concept of quotient space for decomposing FCN. In Section 5, I give a general summary and discuss further research directions.

2. Causal Inference

The Bayesian causal network 1 represents uncertainties with a set of conditional probability functions in the form of a directed acyclic graph (DAG), in which each node represents a concept (or variable), e.g., youth health, smoking, and links (arc/edges) connect some pairs of nodes to represent (possibly causal) relationships [8]. Associated with each node is a probability distribution function given the state of its parent nodes. Traditional expert systems consist of a knowledge base and inference engine. The knowledge base is a set of product rules:

IF condition THEN fact or action

1 In the literature it is also called by other names, such as, causal network, belief network, influence diagram, knowledge map, decision network, to name a few.

© 2002 CFSAT
The inference engine combines rules and input to determine the state of the system and actions. Despite the enormous efforts made in developing useful expert systems over the last three decades, we still face considerable challenges when reason under uncertainty. For instance, when we say “smoking causes lung cancer” or “weaker consumer spending causes lower interest rate” we usually mean that the antecedents may or more likely lead to the consequences. Given uncertainty, we must attach each condition in the rule a measure of certainty which is usually in the form of a probability distribution function.

A Bayesian causal network contains a set of concepts as its nodes, such as inflation, unemployment, and a set of directed links connecting pairs of nodes. Associated with each node is a conditional probability distribution function. The influence of a node to another, say, A to B is represented by a directed link. Figure 1 shows a simple example in which the three concepts influence each other: A has an influence on B which in turn has an influence on C; conversely, the evidence on C will influence the certainty on A through B; or given an evidence on B, we cannot infer C from A and vise versa, which means that A and C are independent given B or the path has been blocked by the evidence on B. For instance, let A stand for smoking, B for lung cancer, and C for hospitalization. If we know that Alice has been smoking for quite some time and also we know her hereditary condition, we may infer that it is likely that she will become a victim of lung cancer, which may lead to hospitalization. On the other hand, if we know that Alice has been hospitalized we may infer that Alice may have lung cancer which may be a result of her smoking problem. However, smoking and being hospitalized become independent, if we known whether Alice has suffered from lung cancer; that is, we have no way of establishing the relationship between smoking and hospitalization.

Figure 2(a) shows another example. In this case A will be the cause for both B and C. On the other hand A and C may influence each other through A. We may say that computer virus (A) may cause files to be erased (B) and computer to crash (C). If we know nothing about A, realizing that some files are missing from the hard drive may lead us to think that the computer might have been attacked by a computer virus; and in turn we may also speculate that some computers on the LAN might have crashed. In this case B and C are dependent. On the other hand, if there was no virus attack, the fact that some files are missing from the hard drive would not change our expectation of computer virus attack. Therefore, “missing files” is not related to computer crash; that is, B and C are independent.

Very often we have to deal with multiple factors that may be the causes to a certain problem. Figure 2(b) illustrates such a case. As it shows, both B and C have a direct influence on A. For the sake of discussion, let’s consider a simple economy case. High inflation (B) or high interest rate (C) can slow down economy growth (A). If we know nothing about the economy growth, we cannot tell what inflation or interest rate might be; that is, they are independent. If we know only that the growth of the economy is slower, we may infer with some certainty that the inflation rate or interest rate has hiked. Further, if we know that indeed the interest rate has hiked; which may also lower our belief of higher inflation rate as the cause; that is, B and C become dependent. This type of reasoning is called explaining away [8] which is very common in human reasoning process that involves multiple causes.

2.1 D-Separation

The Bayesian causal network (BCN) is a powerful framework for inferencing in uncertainty and unexplained exceptions. BCNs describe conditional independence among subsets of variables (concepts) and allow combining prior knowledge about (in)dependencies among variables with observed data. As the above examples show, causes/evidences can traverse the Bayesian causal network in different ways. Given evidence on certain variables (nodes) and certain connections conditional relationships can change between independence and dependence. In BCN, to perform probability inference for query q given n variables, \( v_1, v_2, \ldots, v_n \) that are related to q, we must know \( 2^n \) conditional probability distributions, \( P(q|v_1 \land v_2 \land \ldots \land v_n), \quad P(q|v_1 \land v_2 \land \ldots \land \neg v_n), \ldots, \)

\( P(q|\neg v_1 \land \neg v_2 \land \ldots \land \neg v_n) \). For large systems, this can be intractable. Furthermore, most of the conditional probabilities are irrelevant to the inference. Therefore it
is necessary to reduce the number of probabilities. Fortunately, such a criterion is available which is called direction-dependent separation or simply, d-separation theorem [4]. Basically, this theorem says that if every undirected path from a node in set \( V \) to a node in \( U \) is d-separated by a set of nodes \( E \), then \( V \) and \( U \) are conditionally independent given \( E \). A set of nodes \( E \) d-separates two sets of nodes \( V \) and \( U \) if all paths between \( V \) and \( U \) are blocked by an intermediate node \( F \) given \( E \). A path is blocked if there is a node \( F \) on the path for which either of following two conditions holds:

1. the path is serial (see Figure 1) or diverging (see Figure 2(a))
2. the path is converging (see Figure 2(b)) and neither \( F \) nor any of \( F \)'s descendants is in \( E \).

Figure 3 shows an interesting example for car electric and engine system [9]. Let's look at the relationship between Radio and Gas. We can see from Figure 3 that between Radio and Gas there are four nodes, Battery, Ignition, Starts, and Moves. Given evidence about the (any) state of Ignition, we have a serial path between Radio and Gas: from Battery to Starts, the first condition in the d-separation criterion tells us that the two nodes (variables) Radio and Gas are independent given Ignition. This is correct, since given Ignition the state of Radio will have no influence on the state of Gas and vice versa. Now let's look at the path from Radio to Battery to Ignition to Gas. If we know the Battery is fully charged or otherwise, we have a diverging network similar to Figure 2(a). The criterion tells us again that Radio and Gas are independent, which is correct.

Finally, we look at Starts. In this case we have a converging path between Radio and Gas with Starts as its intermediate node. From the second condition in the criterion, we can conclude that the path is not d-separated. If no evidence about Starts at all, Radio and Gas are independent. However, the situation changes, if car starts and radio works: It increases our belief that the car must be out of gas. That is, Radio and Gas now become dependent. In this case the relationship between Radio and Gas is related to the state of Starts. This is different from the other two cases in which no matter what evidence is given the paths are blocked, hence d-separated.

Indeed, in Bayesian networks we can use the d-separation criterion to read off conditional independences which is perhaps one of the most important features of Bayesian networks.

Over the last 15 years many inference systems have been developed using Bayesian causal networks. However, it has been proved that the exact probabilistic inference using an arbitrary Bayesian causal network is NP-hard [2,3]. Recently researchers have developed machine learning techniques for constructing Bayesian causal networks [5].

Furthermore, for BCNs there is no mechanism available to handle feedback cycles. In dynamic intelligent systems, however, feedback is one of the most important capabilities that enables the system to adjust (adapt) itself in response to the changing environment and the information about the given goals and actual outcomes. Although researchers have attempted to bring dynamics into Bayesian causal networks, they have reported very little progress that is useful in system design.

Bayesian inference is a tool for decision-support and causal discovery in an environment of uncertainty and incomplete information. Since it is based on traditional probability theory, it is difficult to handle such uncertainties as vagueness, ambiguity, and granularity. In such a framework, events in a set are considered all equal and assigned the same binary value: yes or no. This, however, has very little relevance to most real-world problems: We need an alternative.

### 3. Fuzzy Cognitive Maps

Fuzzy cognitive maps (FCM) provide a flexible and more realistic scheme for causal reasoning [6]. In FCM we can use fuzzy sets to represent degrees of causality between events/objects, which enables us to handle uncertainty more effectively. The FCM encodes rules in its networked structure in which all concepts are causally connected. Concepts are activated according to a set of initial conditions and to the underlying dynamics in the FCM. This results in causal inference patterns. As a consequence, with FCM we can represent knowledge and perform inference with greater flexibility than traditional rule-based expert systems or Bayesian causal networks. Furthermore, with FCM we can model different types of uncertainties effectively and combine readily several FCMs into one FCM that takes the knowledge from different experts into consideration.
In real-world applications, FCMs are extremely difficult to analyze due to their large sizes and complex inter-connections. Systematic and theoretical approaches are needed for the analysis and design of fuzzy cognitive maps.

In this paper I will present two major decomposition theories: (1) causal modules and (2) quotient fuzzy causal networks (q-FCNs). A general FCN can be divided into causal modules. The dynamics of a module can be determined by key vertices in the FCN. Based on the concepts of causal modules and key vertices, I propose recursive formulas to describe the dynamics of the key vertices. In q-FCN, the idea is to partition the vertices of an FCN into several blocks according to an equivalence relation on the set of vertices of the FCN. A q-FCN can then be built relative to this partition. Topologically, the vertices of a quotient FCN are the blocks of the partition. One block is connected to another by a directed arc if and only if there is at least one directed arc of the original FCN from a vertex in the first block to a vertex in the second block. The q-FCN shows the causal relationships and the effects between various blocks of the partition. In this way, q-FCN is able to represent global and local information, and their relationships. Both approaches decompose the original FCN into smaller, more manageable parts.

3.1 State Space in FCNs

The FCN is a digraph in which nodes represent concepts and arcs between the nodes indicate causal relationships between the concepts. The connectivity of the FCN can be conveniently represented by an adjacency matrix

\[ W = \begin{bmatrix}
    \cdots & \cdots & \cdots \\
    \cdots & w_{ij} & \cdots \\
    \cdots & \cdots & \cdots 
\end{bmatrix}, \]

where \( w_{ij} \) is the value of the arc from node \( j \) to node \( i \), i.e., value of arc \( a_{ji} \).

Based on causal inputs, the concepts in the FCN can determine their states. This can be determined simply by a threshold [6]. In general, we may define a vertex function as follows.

**Definition 1** A textit function \( f_{r,v_i} \) is defined as:

\[
x_i = f_{r,v_i}(\mu) = \begin{cases} 
1 & \mu > \mu_* \\
0 & \mu \leq \mu_* 
\end{cases}
\]

where \( \mu \) is the total input of \( v_i \), i.e., \( \mu = \sum_k w_{ik} \cdot x_k \).

Usually, if \( T = 0 \), \( f_{r,v_i} \) is denoted as \( f_{v_i} \), or simply, \( f_i \). For the sake of simplicity and without loss of generality, throughout this paper we assume \( T = 0 \) unless specified otherwise.

Given this definition, an FCN can be defined as a weighted digraph with vertex functions. \( \Omega \) denotes an FCN, \( v(\Omega) \), or simply, \( v \) stands for vertex (or concept) of \( \Omega \); \( V(\Omega) \) represents the set containing all the vertices of \( \Omega \); \( x(\Omega) \) or \( x \) is the state of \( v(\Omega) \); \( \phi = (x_1, \ldots, x_n)^T \) denotes the state of \( \Omega \), where \( x_i \) is the state of vertex \( v_i \) and \( n \) is the number of vertices of \( \Omega \); \( a(\Omega) \) or \( a \) stands for an arc in \( \Omega \); \( A(\Omega) \) represents the set containing all the arcs of \( \Omega \); \( v^0(a(\Omega)) \) is the start vertex of \( a(\Omega) \) and \( v^l(a(\Omega)) \) is the end vertex of \( a(\Omega) \).

As every vertex may take a value in \( \{0,1\} \), the state space of the FCN is \( \{0,1\}^n \), denoted by \( X^0(\Omega) \) or \( X^0 \). If the state of \( \Omega \) is \( \phi \) after \( k \) inference steps from an initial state \( \phi_0 \), we say that \( \phi \) can be reached from \( \phi_0 \), or \( \Omega \) can reach \( \phi \) after \( k \) inference steps. Although the initial state \( \phi_0 \) of the FCN may be any state, i.e., \( \phi_0 \in X^0 = \{0,1\}^n \), some states can never be reached no matter which initial state it is set to. For example, state \( (\star \star 1\ 1)^T \) cannot be reached in the FCN shown in Figure 4, where \( \star \) means that it can be any value in \( \{0,1\} \).

**Figure 4. State Space of Fuzzy Causal Network**

We define \( X(\Omega) \) or \( X \) as the reachable state set of \( \Omega \) which contains all states that \( \Omega \) can reach. \( X^\infty(\Omega) \) or \( X^\infty \) is defined as the state set of the states which can be reached by \( \Omega \) after \( 2^n \) inference steps. We have

\[ X^\infty(\Omega) \subset X(\Omega) \subset X^0(\Omega) \]

It is easy to see which state can be reached by the FCN in Figure 4. However, it will be difficult if the FCN contains a large number of concepts and complex connections.
Clearly, if a state \( \phi^* \in X(\Omega) \) there exists a state \( \phi_0 \) such that \( \phi^* \) can be reached in one step inference if \( \phi_0 \) is set as the initial state [7]. We may develop an algorithm to determine whether a state can be reached from a particular initial state.

### 3.2 Causal Module of FCN

A general FCN can be divided into several basic modules, which will be explicitly defined below. Every causal module is a smaller FCN. Vertices (or concepts) of a causal module infer each other and are closely connected. Basic FCN modules are the minimum FCN entities that cannot be divided further.

An FCN \( \Omega \) is "divided" as FCN \( \Omega_1 \) and FCN \( \Omega_2 \) if

1) \( V(\Omega) = V(\Omega_1) \cup V(\Omega_2) \),
2) \( A(\Omega) = A(\Omega_1) \cup A(\Omega_2) \cup B(\Omega_1, \Omega_2) \),

where

\[ B(\Omega_1, \Omega_2) = \{ a(\Omega) \mid V^I(a(\Omega)) \in V(\Omega_1), \quad V^O(a(\Omega)) \in V(\Omega_2) \}, \]

or

\[ V^I(a(\Omega)) \in V(\Omega_2), \quad V^O(a(\Omega)) \in V(\Omega_1), \]

\[ V(\Omega_1) \cap V(\Omega_2) = \emptyset, \]

\[ A(\Omega_1) \cap A(\Omega_2) = \emptyset, \]

\[ A(\Omega_1) \cap B(\Omega_1, \Omega_2) = \emptyset, \]

\[ A(\Omega_2) \cap B(\Omega_1, \Omega_2) = \emptyset. \]

This operation is denoted as

\[ \Omega = \Omega_1 \oplus \Omega_2. \]

Particularly, we consider the causal relationships in subgraphs:

\[ B(\Omega_1, \Omega_2) = \{ a(\Omega) \mid V^I(a(\Omega)) \in V(\Omega_1), \quad V^O(a(\Omega)) \in V(\Omega_2) \}, \]

this case is described as "\( \Omega_2 \) is caused by \( \Omega_1 \)" or "\( \Omega_1 \) causes \( \Omega_2 \)". Such a division is called a regular division, denoted as

\[ \Omega = \Omega_1 \oplus \Omega_2. \]

**Definition 2** An FCN containing no circle is called a simple FCN.

A simple FCN has no feedback mechanism and its inference pattern is usually trivial.

**Definition 3** A basic FCN is an FCN containing at least one circle, but cannot be regularly divided into two FCNs, both of which contain at least one circle.

From the Definitions 2 and 3, we can see that an FCN containing circles can always be regularly divided into basic FCNs. In general, the basic FCNs of \( \Omega \) can be ordered concatenately as follows.

\[ \Omega = \Omega_1 \oplus (\Omega_2 \oplus \Omega_3 \oplus \ldots). \]

We establish a formal result with the following theorem.

**Theorem 1** Suppose FCN \( \Omega \) can be regularly divided into \( m \) basic FCNs, then

\[ \Omega = (\bigcup_{i=1}^{m} \Omega_i) \cup (\bigcup_{i=2}^{m} \bigcup_{j=1}^{i} B(\Omega_j, \Omega_i)). \]

The inference pattern of a basic FCN \( \Omega_i \) is determined by its input (external) and initial state. The inputs of \( \Omega_i \) can be determined once the inference pattern of \( \Omega_k \), \( k \neq i \) are known. Subsequently, the inference pattern of \( \Omega_i \) can be analyzed. If we know the inference patterns of the basic FCNs individually, we will be able to obtain the inference pattern of the entire FCN, because the basic FCNs collectively contain all the concepts of the FCN:

\[ V(\Omega) = \bigcup_{i=1}^{m} V(\Omega_i). \]

The following theorem determines if an FCN is not a basic FCN.

**Theorem 2** Suppose that FCN \( \Omega \) is input-path standardized and trimmed of affected branches. If a vertex \( v_0 \) of \( \Omega \) has at least two input arcs and \( v_0 \) does not belong to any circle, then \( \Omega \) is not a basic FCN.

Theorem 2 provides not only a rule for determining whether an FCN is a basic FCN or not, but also presents an approach to regularly dividing an FCN if it is not a basic FCN: \( I_e(v_0) \) and \( Out(v_0) \). This is illustrated in Figure 5. The FCN in Figure 5(a) can be regularly divided.

---

3 Interested readers may refer to [7] for the proofs in this section.
divided into FCN in Figure 5(b) and in Figure 5(c), respectively, where Figure 5(b) is $I_n(v_g)$ and Figure 5(c) is $Out(v_g)$.

Figure 5. Regularly divided and Basic FCNs: According to vertex $v_g$, FCN in (a) is divided as $Out(v_g)$ in (b) and $Int(v_g)$ in (c).

3.3. Inference Patterns in Basic FCNs

The inference pattern of an FCN can be obtained by recursively calculating

$$\phi(k + 1) = f(W[\phi(k)]) = [f_j(W[\phi(k)])..., f_n(W[\phi(k)])]^T.$$  

However, since in most real applications the FCN contains a large number of vertices and complicated connections, the state sequence can be very long and difficult to analyze. It will be most useful to draw out properties or recursive formula for the state sequence.

**Proposition 1** If an FCN is a simple FCN, it will become static after $L$ inference iterations unless it has an external input sequence, where $L$ is the length of the longest path of the FCN. The Proof of Proposition 1 is obvious. Consequently, the following is also true.

**Corollary 1** Vertices except the end vertex of an input path will become static after $L$ inference iterations unless it has an external input sequence, where $L$ is the length of the path.

In this section, all FCNs are assumed as basic FCNs, with input paths being standardized and affected branches trimmed. For the sake of simplicity, we assume that all FCNs do not have external input sequences unless they are specifically indicated. FCNs with external input sequences can be analyzed in the similar way.

In our study of the inference pattern of the FCN, we found that some vertices may play more important roles than others. We define these vertices as key vertices. The state of every vertex in the FCN can be determined by the state of key vertices. In the following part of this section, the definition of key vertex is followed by some discussions of the properties of key vertex.

**Definition 4** A vertex is called as a key vertex if

1. It is a common vertex of an input path and a circle, or
2. It is a common vertex of two circles with at least two arcs pointing to it which belongs to the two circles, or
3. It is any vertex on a circle if the circle contains no other key vertices.

**Proposition 2** Every circle contains at least one key vertex.

Given Property 3 in Definition 4, the correctness of Proposition 2 is obvious.

**Lemma 1** If any vertex, $v_o \in V(\mathcal{S})$, is not on an input path, then it is on a circle.

**Lemma 2** Suppose that $v_o$ is not a key vertex and not on an input path, then there is one and only one key vertex (denoted as $v^*$) can affect $v_o$ via a normal path, and there is only one normal path from $v^*$ to $v_o$.

Again, suppose the key-vertex set is $\{v_1, ..., v_f\}$. If $v_j (1 \leq j \leq r)$ satisfies (1) of Definition 4, it is the end vertex of an input branch, denote $v'(j)$ as the input vertex. Denote the normal paths from $v_j$ to $v_j$ as $P_0^j, P_1^j, ..., P_{r(i,j)}^j$, where $r(i,j)$ is the number of normal paths from $v_j$ to $v_j$. $P_0^j = P_{r(i,j)}^j$ means that there is no normal path from $v_j$ to $v_j$ and $f_{P_0^j} = 0$.

**Theorem 3** If $v_j (1 \leq j \leq r)$ is not the end of an input branch,

$$x_j(l) = f_{T_j,x_j} \left( \sum_{i=1}^{s} f_{P_i^j} \{ x_i(l-L_{P_i^j}) \} \right).$$  

If $v_j$ is the end of an input branch,

$$x_j(l) = f_{T_j,x_j} \left( \sum_{i=1}^{s} f_{P_i^j} \{ x_i(k-L_{P_i^j}) \} + f_{P_{r(i,j)}^j} \{ x_{r(i,j)}(l-L(P_{r(i,j)}^j)) \} \right).$$

Therefore, the inference pattern of the FCN is can be determined by the recursive formula in terms of key vertices. After the states of key vertices are determined, the states of the remaining vertices can be determined by states of key vertices as follows. In turn, the state of the entire FCN is also determined.

**Theorem 4** Suppose that $K_\nu(\mathcal{S})$ is the key vertex set of $\mathcal{S}$, $I_\nu(\mathcal{S})$ is the input vertex set of $\mathcal{S}$, $v_o \in V(\mathcal{S})$, and $v_0 \not\in I_\nu(\mathcal{S})$. There exists one and only one vertex $x^*$, $x^* \in K_\nu(\mathcal{S})$ or $x^* \in I_\nu(\mathcal{S})$ such that there exists a path, $P^*(v^*, v_0)$ from $v^*$ to $v_0$ via no vertices in $K_\nu(\mathcal{S})$.

**Theorem 5** $P^*(v^*, v_0)$ is a normal path.

As $P^*(v^*, v_0)$ is a normal path,

$$x_0(l) = f_{p^*(v^*, v_0)} \{ x^*(l-L(P^*(v^*, v_0))) \}.$$
Thus the state of the entire FCN is determined. When considering the inference pattern of a general FCN (\(\mathcal{U}\)), we should first regularly divide it into basic FCNs (\(\mathcal{U}_i, 1 \leq i \leq m\)) and then determine the inference pattern one by one according to the causal relationships between of them. For the basic FCN (\(\mathcal{U}_i\)), the external input should be divided according to the inference pattern of \(\mathcal{U}_j, 1 \leq j \neq i\). Then the input paths should be standardized. After this process, we delete all the affected branches to simplify the FCN (\(\mathcal{U}_i\)) for further analysis.

From the simplified FCN, we need to construct the key-vertex set. If the basic FCN contains only one circle, and every vertex on the circle has only one input arc, then the key-vertex set contains only one vertex. It can be any vertex on the circle according to Definition 3. If the basic FCN contains more than one circle, the key vertices are the circle vertices that have at least two input arcs. This can be judged from \(W\). If the \(i\)th column of \(W\) has at least two non-zero elements, \(v_i\) has at least two input arcs.

### 4. Quotient FCNs

An FCN can be considered as a collection of vertex set and arcs that are highly organized according to functionalities and causal relationships. Therefore, it is possible to partition the vertex set and hierarchically re-model the FCN into a new FCN which is functionally equivalent to the original FCN and easier for analysis and design. In this section, I will briefly present another decomposition theory based on the theory of quotient space. First, let's give some definitions. (Interested readers may consult [10] for more details.)

#### 4.1 Definitions and Theory of q-FCN

**Definition 5** Given two sets \(S\) and \(T\), a binary relation \(\rho\) between \(S\) and \(T\) is a subset of the Cartesian product \(S \times T\). If \((x, y) \in \rho\), then we say that \(x\) and \(y\) have the binary relation \(\rho\), and we denote this fact by \(x \rho y\). In particular, for a set \(S\), a binary relation on \(S^2\) (the set of ordered pairs of elements of \(S\)) is called a binary relation on \(S\).

**Definition 6** Let \(\rho\) be a binary relation on a set \(S\). Then \(\rho\) is reflexive ⇔ \((\forall x)(x \in S \Rightarrow (x, x) \in \rho)\)

\(\rho\) is symmetric ⇔ \((\forall x)(\forall y)(x \in S, y \in S, (x, y) \in \rho \Rightarrow (y, x) \in \rho)\)

\(\rho\) is transitive

\[\Rightarrow (\forall x)(\forall y)(\forall z)(x \in S, y \in S, z(S)(x, y) \in \rho, (y, z) \in \rho \Rightarrow (x, z) \in \rho)\]

A binary relation \(\rho\) which is reflexive, symmetric and transitive is called an equivalence relation.

**Definition 7** A partition of a set \(S\) is a family \(B = \{B_1, ..., B_k\}\) of nonempty subsets \(B_1, ..., B_k\) of \(S\) satisfying the following conditions:

(a) \(B_1 \cup B_2 \cup ... \cup B_k = S\);

(b) \(B_i \cap B_j = \emptyset\) if \(i \neq j\).

In this case each \(B_i\) is called a block of the partition \(B\). A quotient FCN is constructed according to a partition of vertices.

**Definition 8** Suppose \(\mathcal{U} = (V, E)\) is an FCN with \(n\) vertices, and \(B = \{V_1, ..., V_k\}\) is a partition of the vertex set \(V\). Define a new FCN with vertices the blocks of \(B\) such that \((V_i, V_j)\) is an arc if and only if there exists at least one arc of \(\mathcal{U}\) initiating at a vertex in \(V_i\) and terminating at a vertex in \(V_j\). We call this new FCN the quotient fuzzy causal map (q-FCN) of \(\mathcal{U}\) relative to the partition \(B\), or simply a q-FCN, and denote it by \(\mathcal{U}_B = (V_B, E_B)\).

**Definition 9** Let \(\mathcal{U}\) and \(B\) be as in Definition 8. For each \(V_i \in B\), let \(E_i\) denote the set of arcs of \(\mathcal{U}\) with both end-vertices in \(V_i\). We call \(\mathcal{U}_i = (V_i, E_i)\) the sectional fuzzy causal map (s-FCN) of \(\mathcal{U}\) on \(V_i\), or simply the s-FCN on \(V_i\). The strength with which one vertex in \(V_i\) influences another vertex in \(V_i\) is defined to be the same as that in \(\mathcal{U}\).

**Theorem 6** Suppose \(\mathcal{U} = (V, E)\) is an FCN, and \(B = \{V_1, ..., V_k\}\) is a partition of the vertex set \(V\). Let \(\mathcal{U}_B = (V_B, E_B)\) be the quotient FCN associated with \(B\).

Then

\[\mathcal{U}_B = \left( \bigcup_{i=1}^{k} \mathcal{U}_i \right) \cup \left( \bigcup_{j=1}^{k} \bigcup_{i=1}^{j} B(\mathcal{U}_j, \mathcal{U}_i) \right),\]

where \(\mathcal{U}_i = (V_i, E_i)\) is the sectional FCN of \(\mathcal{U}\) induced on \(V_i\) and

\[B(\mathcal{U}_j, \mathcal{U}_i) = \begin{cases} \{(v_i, v_j) | v_i \in V_i, v_j \in V_j \text{ if } i \neq j\} & \text{if } i = j \\ \emptyset & \text{if } i \neq j. \end{cases}\]

The proof of this theorem is similar to that of Theorem 2 in [7]. Furthermore, we can see that

\[V(\mathcal{U}_i) \cap V(\mathcal{U}_j) = \emptyset, \quad i \neq j, \quad E(\mathcal{U}_i) \cap V(\mathcal{U}_j) = \emptyset, \quad i \neq j,\]
which show that the analysis of the original FCN can be reduced to that of the quotient FCN and sectional FCNs. As usual, in order to perform inference, for each, we need to define the states of vertices and the strength with which each vertex influences its neighboring vertices. For the sake of simplicity and without loss of generality, we consider only binary vertex states.

**Definition 10** Let \( \mathcal{U} \) and \( B \) be as in Definition 8, and \( \mathcal{U}_B = (V_B, E_B) \) the quotient FCN associated with \( B \). Let \( x_i^B(t) = x_i^B(\mathcal{U}_B) \) stand for the state of the vertex \( V_i \) of \( \mathcal{U}_B \) at time \( t \) and \( \phi^B(t) = (x_1^B(t),...,x_k^B(t)) \) for the state of \( \mathcal{U}_B \) at \( t \). We define \( x_i^B(t) \) such that \( x_i^B(t) = 1 \) if and only if there is at least one vertex \( V_j \in V_i \) with \( x_j(t) = 1 \) and at least one arc \((V_j, V_i)\) direct to another block \( V_j, (1 \leq j \leq k, j \neq i) \) in \( B \), where \( x_i(t) \) is the state of \( V_i \) at \( t \). In other words, we define

\[
x_i^B(t) = \max_{1 \leq j \leq k, j \neq i} x_j(t), \text{ for each } i = 1,...,k.
\]

This definition says that the state value \( x_i^B(t) \) of vertex \( V_i \) at time \( t \) is initially determined by the state values of the vertices in \( V_i \), with at least one directed arc from these vertices to the vertices in the other blocks.

To calculate the weight \( f_{ij} \) on the arc \((V_i, V_j)\) of \( \mathcal{U}_B \), we may use t-norm and t-conorm, which generate the intersection and union operations.

### 4.2 Causal Algebra in Quotient FCNs

As we discussed before, FCN is an inference network. Therefore a quotient FCN must calculate accurately the inference pattern. In this section, we present the causal algebra for computing quotient FCN.

Let \( \mathcal{U}, w_{\mathcal{U}}, \phi_{\mathcal{U}} \) be an FCN. Let \( B = \{V_1,...,V_k\} \) be a partition of the vertex set \( \mathcal{U} \). Let the triple \( (\mathcal{U}_B, w_{\mathcal{U}_B}, \phi_{\mathcal{U}_B}) \) be the q-FCN of \( \mathcal{U} \) associated with \( B \). Recall that \( \mathcal{U}_B = (V_B, E_B) \), \( w_{\mathcal{U}_B} = E(\mathcal{U}_B) \rightarrow [-1,1] \), \((V_i, V_j) \mapsto f_{ij} \) defined on the arcs of \( \mathcal{U}_B \), where \( f_{ij} \) is defined in Definition 11, and \( \phi_{\mathcal{U}_B} \) is the state of q-FCN at \( t \) represented by the vector function:

\[
\phi_{\mathcal{U}_B}(t) = (x_1^B(t),...,x_k^B(t)).
\]

Consequently, we can express the adjacency matrix of a q-FCN as follows:

\[
W_{\mathcal{U}_B} = \begin{bmatrix}
\cdots & \cdots & \cdots \\
\cdots & f_{ij} & \cdots \\
\cdots & \cdots & \cdots
\end{bmatrix}
\]

**Definition 11** Let be an FCN. Let \( B = \{V_1,...,V_k\} \) be a partition of the vertex set \( \mathcal{U} \), and let \( (\mathcal{U}_B, w_{\mathcal{U}_B}, \phi_{\mathcal{U}_B}) \) be a quotient FCN, then we define the threshold \( T_i \) at vertex \( V_i \) as the maximum of all the thresholds consisting of \( T_j \), such that \( V_j \in V_i \), and for such a vertex \( V_j \) there is at least one arc \((V_j, V_i)\) direct to another block \( V_j, (1 \leq j \leq k, j \neq i) \) associated with \( B \). In other words, we can define the threshold \( T_i \) at vertex \( V_i \) in a quotient FCN as follows.

\[
T_i = \max_{v \in V_i} \{ T_{ij} \},
\]

Based on Definition 11 we can determine the state of \( \phi_{\mathcal{U}_B}(t) \) of \( \mathcal{U}_B \) at \( t \) by an initial condition and thresholds \( T_i \) at vertex \( V_i, i \leq i \leq k \). When \( \phi_{\mathcal{U}_B}(t) \) receives a series of external inputs, its next state \( \phi_{\mathcal{U}_B}(t+1) \) is computed according to the following formula,

\[
\phi_{\mathcal{U}_B}(t+1) = f_T(\phi_{\mathcal{U}_B}(t)) \times W_{\mathcal{U}_B},
\]

where

\[
T = (T_1,...,T_k), \phi_{\mathcal{U}_B}(t) \times W_{\mathcal{U}_B} = (\mu_1,...,\mu_k),
\]

\[
f_T(\phi_{\mathcal{U}_B}(t)) \times W_{\mathcal{U}_B} = (f_T(\mu_1),...,f_T(\mu_k)),
\]

where \( \mu_i \) is the total inputs of \( V_i \),

\[
\mu_i = \mu_{\mathcal{U}_B}(V_i) = \sum_{j=1}^{k} f_{ij} \cdot x_j^B(t)
\]

and \( f_T(\mu_i) \) is the vertex function at \( V_i \) defined as follows.

**Definition 12** Given a threshold \( T_i \) for the \( i \)th vertex \( V_i \), the vertex function \( f_{T_i} \) of \( V_i \) is defined by

\[
f_{T_i}(\mu_i) = \begin{cases} 1, & \text{if } \mu_i \geq T_i, \\ 0, & \text{if } \mu_i < T_i. \end{cases}
\]

**Theorem 7** Let \( (\mathcal{U}_B, w_{\mathcal{U}_B}, \phi_{\mathcal{U}_B}) \) be an FCN, \( B = \{V_1,...,V_k\} \) a partition of the vertex set of \( V \), \((\mathcal{U}_B, w_{\mathcal{U}_B}, \phi_{\mathcal{U}_B}) \) the q-FCN associated with partition \( B \), and \( T_i \) a threshold at vertex \( V_i \) of the quotient FCN as in Definition 11. The following properties hold at any \( T_i \):

1. If the vertex \( V_i \) in the q-FCN is active, then at least exist one vertex \( v_i \) of \( V_i \) in the FCN, such that
Proposition 3 Let $P = P(U,V)$ denote a directed from vertex $U$ to vertex $V$ in the q-FCN. Let $I_{(U,V)}(t)$ denote the indirect effect with which vertex $U$ influences vertex $V$ via $P$. Then

$$I_p(t) = \prod_{(Y,Z) \in E(P)} x^B_i(t) \cdot f_{YZ}$$

(11)

where $E(P)$ is the set of arcs on $P$, $x^B_i(t)$ is the state of vertex $Y$ at time $t$, and $f_{YZ}$ is the weight associated with arc $(Y,Z)$ on $P$.

Proposition 4 Let $N(V) = \{U \mid U \in V_b, (U,V) \in E_b\}$ be the set of vertex $U$ of $V_b$ such that there is an arc from $U$ to $V$, i.e., $N^-(v)$ is the in-neighborhood [10] of the vertex $V$ in the q-FCN. For vertices $U$, $V$ of $V_b$, we denote $P(U,V)$ as the set of directed paths from $U$ to $V$. For $U_i \in N^-(v)$, let $P(U,U_i,V)$ denote a directed path from $U$ to $V$ passing the vertex $U_i$, and let $P(U,U_i,V)$ be the set of all such paths. For $P \in P(U,V)$, let $I_p(t)$ denote the indirect effect of $U$ on $V$ at time $t$ via $P$. Let $T_{(U,V)}(t), I_{(U,V)}(t)$ and $I_{n(U,V)}(t)$ be the total, strongest and weakest effect that $U$ influences $V$ via all directed paths from $U$ to $V$, respectively. Then the following properties hold:

(a) If $P = P(U,V)$ consists of the distinct vertices sequences $U = V_1, \ldots, V_t = V$, then the indirect effect of $U$ on $V$ at time $t$ via $P$ can be written as follows:

$$I_p(t) = I_{(V_1,V_2,\ldots,V_t)}(t) = \prod_{i=1}^{t-1} x^B_i(t) \cdot f_{t_i \, \overline{t_i+1}}$$

(b) $I_{(U,V)}(t) = \min_{P \in P(U,V)} I_p(t)$.

(12)

(c) $I_{(U,V)}(t) = \max_{P \in P(U,V)} I_p(t)$.

(13)

d) $I_{(U,V)}(t) = \min_{P \in P(U,V)} I_p(t)$.

(14)

(e) If there is a unique directed path, $P = P(U,V)$, from $U$ to $V$, then

$$T_{(U,V)}(t) = I_{(U,V)}(t) = I_{n(U,V)}(t) = I_p(t).$$

(15)

The proof of this proposition is similar to that in [7].

Theorem 8 Let $p(U,V)$ be a directed path in the quotient FCN [10]. Let $I_{(U,V)}(t), T_{(U,V)}(t), I_{(U,V)}^*(t)$ and $I_{n(U,V)}^*(t)$ be as in Proposition 4. Then the state of the $I_{(U,V)}(t), I_{(U,V)}^*(t), I_{n(U,V)}^*(t)$ and $T_{(U,V)}(t)$ at time $t$ are also determined by an initial condition and given thresholds $T_i$ at vertex $V_i$, $1 \leq i \leq k$. When $\phi_{0k}(t)$ receives a series of external input sequences, their next states will be updated according to (11, 13, 14) and (15), respectively until static states are reached.

The proof of the theorem is similar to that of Theorem 2.8 in [10].

I have presented a major decomposition theory using the quotient space theory. This theory can be used as an effective tool for the analysis and design of fuzzy causal networks. To summarize, we can implement the proposed decomposition theory recursively as follows:

1. Model the real-world problem in terms of an FCN;
2. Define an appropriate equivalence relation $\rho$ for the vertex set $V$ and then partition $V$ into some blocks $B = \{V_1, \ldots, V_k\}$ according to the equivalence relation;
3. Regard each block $V_i$ as a new vertex, and then construct a q-FCN based on these new vertices;
4. Calculate the vertex state values and the weights of the arcs relative to the quotient FCN. Then analyze the causal inference of the quotient FCN, which can provide the global information about the original FCN;
5. Each block $V_i$ represents a sectional FCN (s-FCN). The s-FCNs collectively maintain the topological structure of the original FCN; whereas the individual s-FCNs provide the local information about the original FCN.

I have presented briefly a major decomposition theory using the quotient space theory. This theory can be used as an effective tool for the analysis and design of fuzzy causal networks.

5. Conclusions

In this paper I have given a brief discussion of the theory of causation, and its implications in decision-support systems. The Bayesian causal network uses conditional probability to model causal relationships and uncertainty of the system. Although the
BCN has received considerable attention in the research community, it still remains an illusive goal for developing robust, functional systems for real-world applications. One of the most difficult problems is that we cannot use BCNs to model vagueness, ambiguity, and natural descriptions that present in virtually all practical applications. Fuzzy causal networks use the theory of causation and fuzzy set and are able to model naturally occurring uncertainties. Indeed, it is a powerful framework for developing decision-support systems. However, for complex problems, it remains difficult to analyze and design an FCN. In this paper, based on our recent research, I have introduced some basic concepts and theory of two major decomposition techniques: module FCNs and q-FCNs.

I have to stress that although Bayesian causal networks and fuzzy causal networks were proposed during the same period (the 1980s), compared to BCNs we have yet to see significant theoretical development for the analysis and design of FCNs. In this area many challenging and interesting problems still remain open.

References


Zhi-Qiang Liu is a Professor with School of Creative Media, City University of Hong Kong and Department of Computer Science and Software Engineering, the University of Melbourne. He received the M.A.Sc. degree in Aerospace Engineering from the Institute for Aerospace Studies, The University of Toronto, and the Ph.D. degree in Electrical Engineering from the University of Alberta, Canada. His interests are scuba diving, neural-fuzzy systems, classic music, gardening, machine learning, mountain/beach trekking, human-media systems, computer vision, serving the community, mobile computing, and computer networks.