第 27 卷　第 4 期　　自 动 化 学 报　　Vol.27,No.4
2001 年 7 月　　ACTA AUTOMATICA SINICA　　July, 2001

Causation, Bayesian Networks, and Cognitive Maps

LIU Zhi-Qiang

(*Intelligent Systems Lab.(ISL), Department of Computer Science and Software Engineering,*
University of Melbourne, VIC 3010, *Australia*)
(E-mail: z.liu@computer.org)

Abstract Causation plays a critical role in many predictive and inference tasks. Bayesian networks (BNs) have been used to construct inference systems for diagnostics and decision making. More recently, fuzzy cognitive maps (FCMs) have gained considerable attention and offer an alternative framework for representing structured human knowledge and causal inference. In this paper I briefly introduce Bayesian networks and cognitive networks and their causal inference processes in intelligent systems.

Key words Bayesian networks, fuzzy cognitive map, causation, inference, decision making, dynamic systems, intelligent systems, graphs.

1 Introduction

In many real-world applications, we use the available information for making analysis and reaching decisions.Data analysis and decision making processes can be broadly considered as a prediction process. In general there are two types of tasks in such processes, which require different methods:

1) Classification which is concerned with deciding the nature of a particular system given the features, which usually produces *labeled* data;

2) Causal prediction which is concerned with the effect of the *changes* in some features to some other features in the system.

For instance, to classify a car, we may use color, size, and shape as its features. Based on the features, we may reach a conclusion that a particular car in the image is a passenger car or a bus. In many applications, we may use classification to gain new knowledge. For example, in searching for anti-AIDS vaccine, researchers use data from certain population that has exposed to HIV but not developed AIDS to find a procedure to develop effective anti-AIDS vaccine. In such problems the process uses features for prediction and does not concern the effect of the *change* in the feature to the system. Classification has been a major topic in machine learning. Many techniques have been developed for the classification and pattern recognition problem, e.g., neural networks, decision trees, various learning paradigms, such as competitive learning, supervised learning, reinforcement learning, etc.

The second major task is mainly related to Causal Discovery or Causal Inference which is concerned with the *change* of features in the prediction (or inference) process, e.g., the effect of a *new* investment strategy to the profits of the investor; the impact of the *increased* cigarette tax on the health of youth; the influence of *wider* exposure to violent video games on social behavior of high school students. The changes in these examples would directly or indirectly alter some of the features in the data. To know the effect of changes, we must have some mechanisms that can discover the cause and effect relations from the data set. In this paper I will briefly discuss causality and two major approaches to the problem of causal inference: Bayesian networks (BN) and fuzzy cognitive maps (FCM).

Received 20 August, 2000; in revised form 24 September, 2000
收稿日期 2000-08-20　　收修改稿日期 2000-09-24

The Bayesian network[1] is a causal inference model that 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 (edges) connect some pairs of nodes to represent (possibly causal) relationships[1~3]. Associated with each node is a probability distribution function given the state of its parent nodes. Since the early 80s research on Bayesian networks has gained considerable interests. Bayesian networks provide a convenient framework to represent causality, e.g., rain causes wet lawn if we know it has rained. Bayesian networks have found some applications. For instance, Andreassen *et al.* have developed an expert system using the Bayesian network for diagnosing neuromuscular disorders[4]; based on blood types Rasmussen has developed a system, BOBLO, for determining the parentage of cattle[5]; Binford and his colleagues have used the Bayesian network for high-level image analysis[6,7]; and the PATHFINDER system for pathology[8].

About same period when the theory of Bayesian networks was introduced. Kosko[9] proposed the basic structure and inference process of the Fuzzy Cognitive Map which was based on the Cognitive Map developed by Axelrod[10]. The FCM encodes rules in its networked structure in which all concepts are *causally* connected. Rules are fired based on a given set of initial conditions and on the underlying dynamics in the FCM. The result of firing of the concepts represents the causal inference of the FCM. In FCMs we are able to represents all concepts and arcs connecting the concepts in terms of symbols or numerical values. As a consequence, we can represent knowledge and implement inference with greater flexibility. Furthermore, in such a framework it is possible to handle different types of uncertainties effectively and to combine readily several FCMs into one FCM that takes the knowledge from different experts into consideration[11]. The theory of FCMs represents a very promising paradigm for the development of *functional* intelligent systems[12~15].

2 Bayesian Networks

Causality plays an important role in our reasoning process. Let's look at the following two simple examples:

1) "When it rains the lawn will be wet."

2) We know that speeding, illegal parking, drink-drive, etc. violate traffic regulation which can incur a fine by the police. "If John speeds he may get a fine."

A functional intelligent system must have the ability to make decisions based on causal knowledge or causal relationships. Based on causal knowledge we are able to *causally* explain probable outcomes given known relationships between certain actions and consequences, e.g., "smokers are at a higher risk of contracting lung cancer" is based on the probable cause (smoking) of the effect (lung cancer).

2.1 Causal Inference

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*

The inference engine combines rules and input to determine the state of the system and actions[16] Whereas a few impressive rule-based expert systems have been developed, it has been demonstrated that it is difficult to reason under uncertainty. For instance, when we say "smoking causes lung cancer" or "higher inflation figure causes interest rate to rise" we usually mean that the antecedents *may* or *more likely* lead to the consequences. Under such a situation,

[1] This is most commonly used name. In the literature it is also called by different names, such as, causal network, belief network,influence diagram, knowledge map, decision network, to name a few.

we must attach each condition in the rule a measure of certainty which is usually in the form of a probability distribution function. Deterministic rules can also in many cases lead to paradoxic conclusions, for instance, the two plausible premises:

1) My neighbor's roof gets wet whenever mine does.

2) If I hose my roof it will be wet.

Based on these two premises, we can reach the implausible conclusion that my neighbor's roof gets wet whenever I hose mine. To handle such paradoxical conclusions we must specify in the rule for all possible exceptions:

My neighbor's roof gets wet whenever mine does, except when it is covered with plastic, or when my roof is hosed, etc.

Bayesian networks offer a powerful framework that is able to handle uncertainty and to tolerate unexplicated exceptions. Bayesian networks describe conditional independence among *subsets* of variables (concepts) and allow combining prior knowledge about (in) dependencies among variables with observed data. A Bayesian network is a directed graph that contains a set of nodes which are random variables (such as *speeding, illegal parking*), and a set of directed links connecting pairs of nodes. Each node has a conditional probability distribution function. Intuitively, the directed link between two nodes, say, A to B means that A has a direct influence on B. Fig.1 shows a simple example.

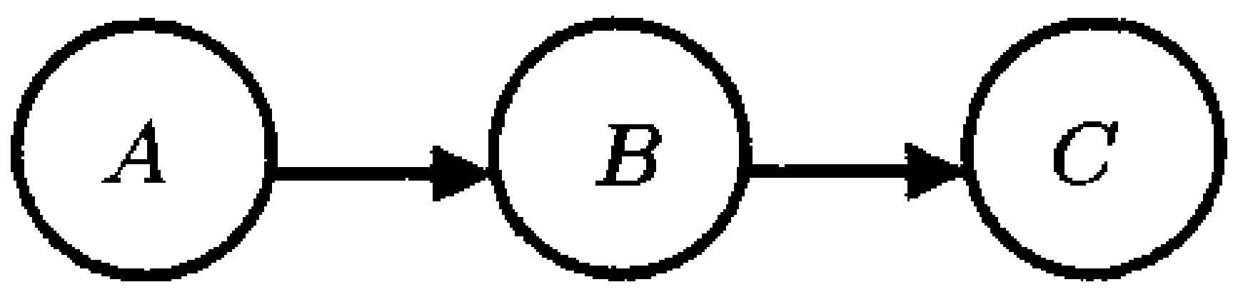
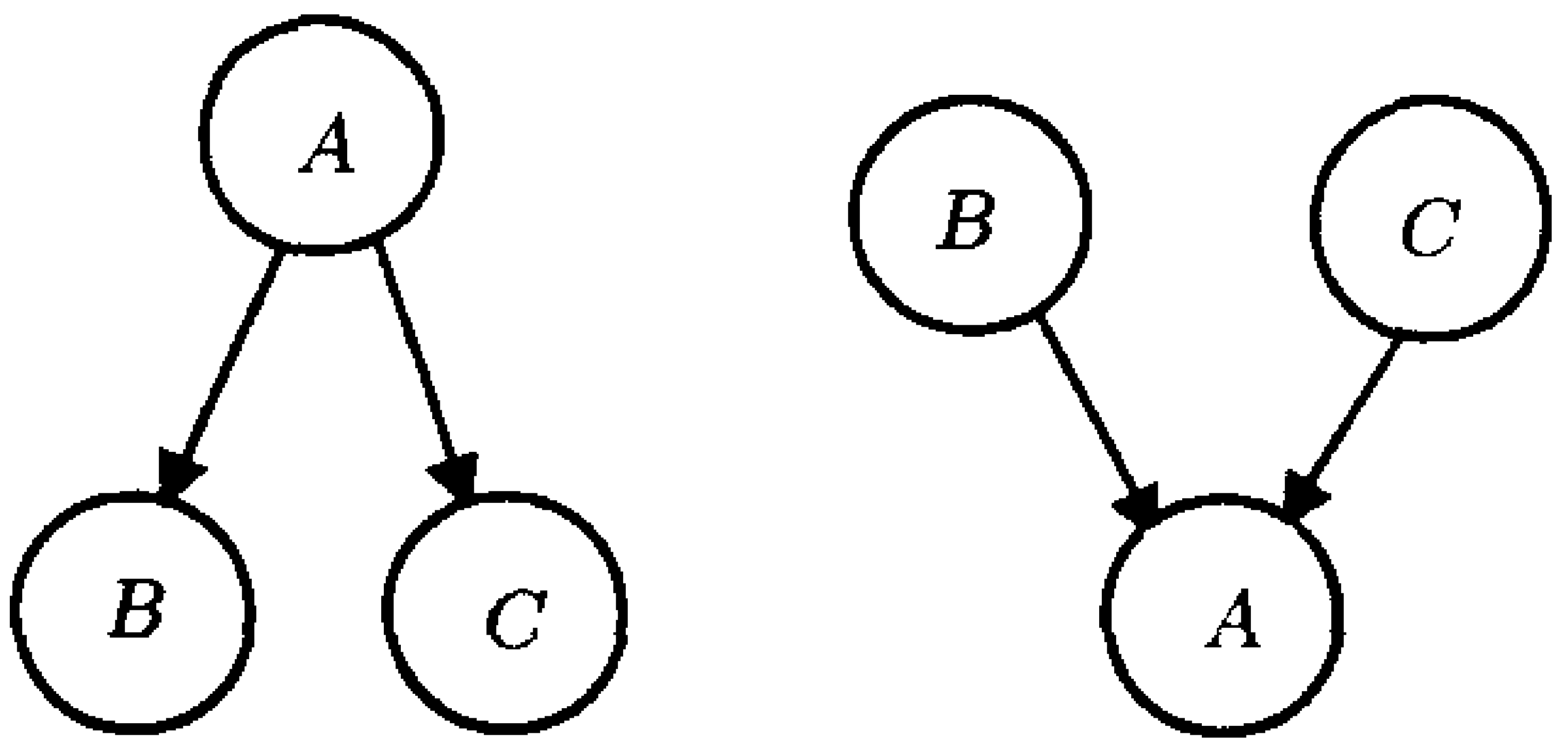


Fig.1 An example of a simple serial connection

In Fig.1 all three variables have influence on each other in different ways: 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 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 *drinking*, B for *traffic accident*,and C for *fine*. If we know that John has drunk, if he also drives he will *likely* get involved in an accident, and consequently get a fine. On the other hand, if we know that John has received a fine we may infer that John may have involved in an accident which may be a result of his drinking problem. However, drinking and getting a fine become independent, if we known whether John has involved in a traffic accident,

Fig.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 drinking (A) may cause traffic accident (B) and fight (C).If we know nothing about A, knowing that John has involved in a traffic accident may trigger us to infer that he might be drinking too much, and in turn we may also speculate that he might also getting into a fight. In this case B and C are dependent. Now, if we were told that John was not drinking, the fact that John has involved in an accident will not change our expectation concerning his drinking problem; consequently, traffic accident has no influence on fight.In this case B and C are independent.



(a) Diverging connections (b) Converging connections

Fig.2 Simple Bayesian networks

In Fig.2 (b), both B and C have a direct influence on A. traffic violation (B) or fight (C) can result in a fine (A). If we know nothing about the fine, traffic violation and fight are independent. If we know only that John has received a fine, we may infer with some certainty that John has committed traffic violation or got into a fight. Now we are told that John has indeed committed traffic violation; our belief in John also got into a fight will reduce; that is, B and C become dependent. This type of reasoning is called *explaining away*[3] which is very common in human reasoning process that involves multiply causes. For instance, when diagnosing a disease the doctor may use the obtained evidences, e.g., a lab report, to eliminate other possible disorders.

2.2 D-Separation

In the above examples we have discussed possible ways evidence can traverse the Bayesian network.We have found that given evidence on certain variables (nodes) and certain connections conditional relationships can change between independence and dependence. In general, in order to perform probability inference for query q given n variables,v_1 v_2 $\cdots$ v_n that are related to q, we must know 2^n conditional probability distributions, $P(q|v_1 \wedge v_2, \wedge \cdots \wedge v_n)$, $P(q|v_1 \wedge v_2 \wedge \cdots \wedge \neg v_n), \cdots$, $P(q|\neg v_1 \wedge \neg v_2 \wedge \cdots \wedge \neg v_n)$.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.

The Bayesian network provides a convenient representational framework. Indeed given a Bayesian network, it will be beneficial if we can find a criterion to systematically test the validity of general conditional independence conditions.For instance, if we are able to determine that in a given network a set of variables **V** is independent (or dependent) of another set of variables **U** given a set of evidence **E**. This will reduce the conditional dependence in general, hence reducing the amount of information needed by a probabilistic inference system.

Fortunately, such a criterion is available which is called *direction-dependent separation* or simply, *d-separation* theorem[3,17]. 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**[18]. 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 Fig.1) or diverging (see Fig.2(a)) if $F \in$ **E**; or

2) the path is converging (see Fig.2(b)) and neither F nor any of F's descendents is in **E**.

Fig.3 shows an interesting example for car electric and engine system[16]. Let's look at the relationship between *Radio* and *Gas*. We can see from Fig.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 vise 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 Fig.2(a).The criterion tells us again that *Radio* and *Gas* are independent, which is correct.

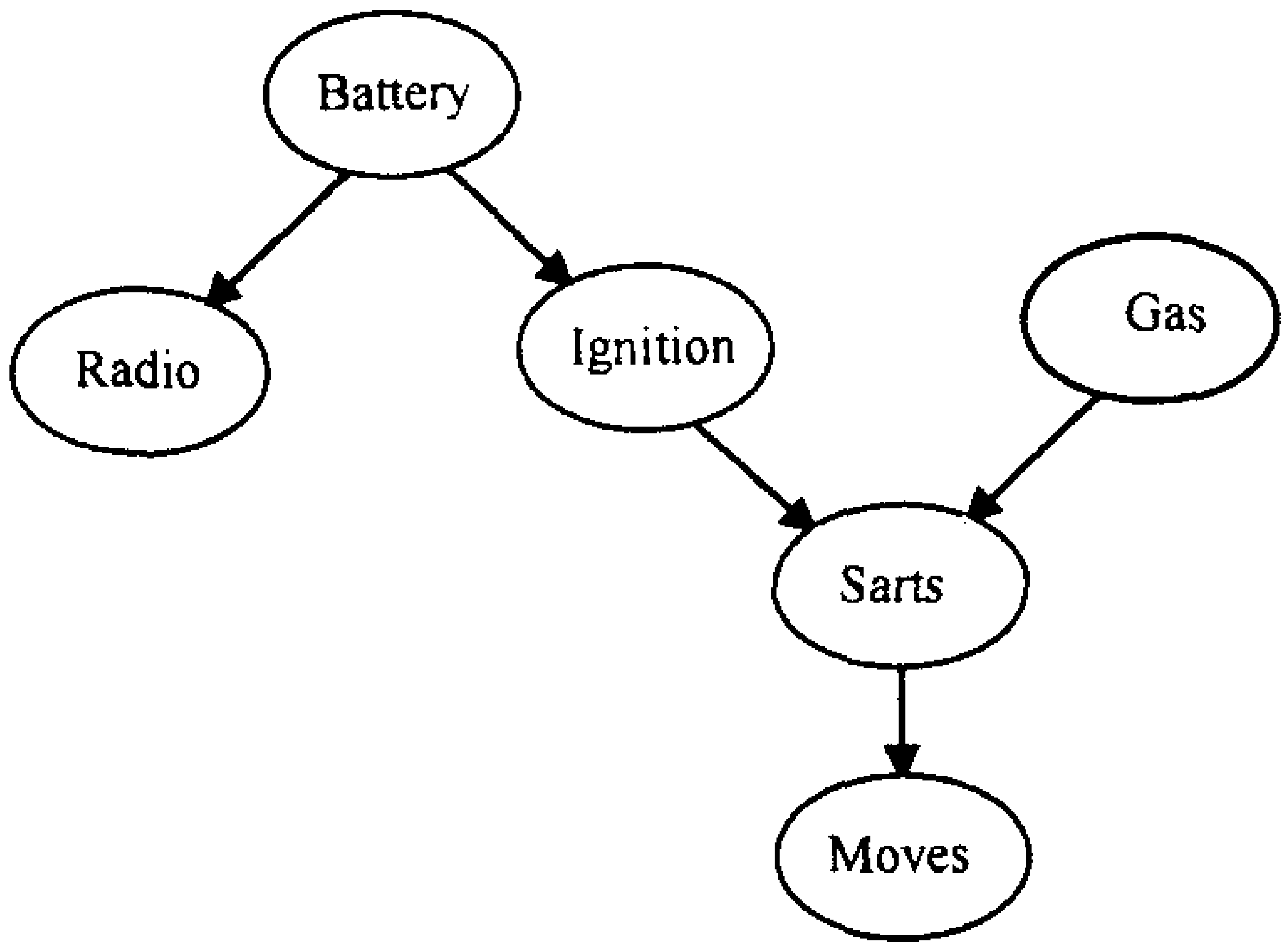


Fig.3　The Bayesian Network for car electric and engine system

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 d-separation to *read off* conditional independencies which is perhaps one of the most important features of Bayesian networks.

2.3 Inference Using Bayesian Networks

Let's consider the following scenario. On one autumn afternoon, Ms. Doubtfire was shopping at the local grocery store when Ms. Gibbons rushed in telling her that she saw smoke from the direction of Ms. Doubtfire's house. Ms. Doubtfire thought it must be the stew that had been cooked for too long and caught fire. On her way running home she recalled that her next door neighbor Mr. Woods burns garden cuttings every autumn. Based on this story we can build a causal network as shown in Fig.4. This network represents the causal knowledge. Fig.4 shows that burning garden cuttings and stew caught fire directly affect the probability of smoke. In the this network we have not directly linked *burn cutting* and *stew on fire* with the action of Ms. Gibbons and indeed Ms. Doubtfire's.

Given the relationships described in Fig.4, we can assign *conditional* probability function for each node (or concept). Let B represent the node, *burning garden cuttings*, S stand for *stew on fire*,SM for *smoke*, G for *Ms. Gibbons tells*, and D for *Ms. Doubtfire runs home*. Given either B or S, the conditional probability for smoke SM, may be written as $P(SM|B,S)$. The conditional probability table for smoke SM may look like the following:

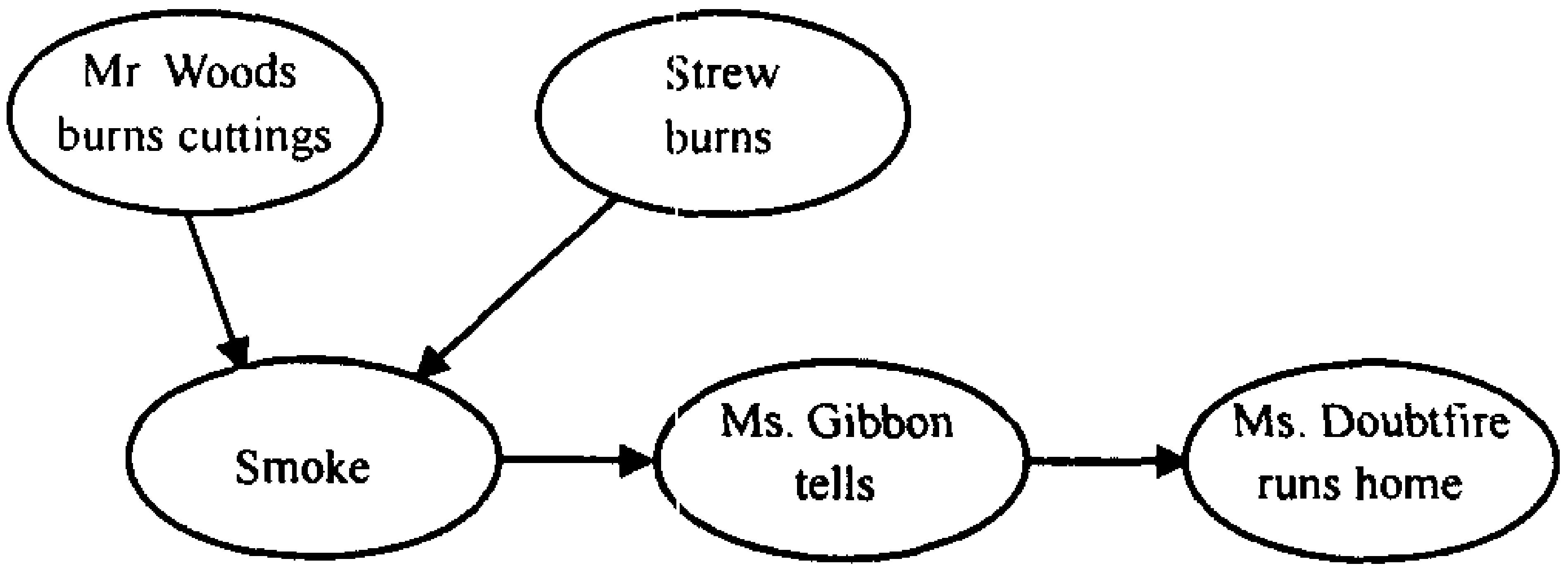


Fig.4 Should Ms. Doubtfire be rushing home?

		$P(SM\|B,S)$	
B	S	*True*	*False*
True	*True*	0.95	0.05
True	*False*	0.95	0.05
False	*True*	0.34	0.66
False	*False*	0.002	0.998

To perform inference in the Bayesian network, we need to computer the posterior probability distributions for a set of query variables on the knowledge of evidence variables, $P(\text{query}|\text{evidences})$. In Fig.4, Mr. Woods burns garden cuttings and stew over cooked are two query variables, Ms. Gibbons' telling could be the evidence variable[1]. For this example based on the d-separation criterion we need only the following conditional probabilities (instead of 32): $P(G|SM)$, $P(D|G)$, and $P(SM,B,S)$ which is

$$P(SM,B,S)=P(SM|B,S)P(B,S)=P(SM|B,S)P(B)P(S),$$

because B and S are independent if we know nothing about SM.

[1] Although this example looks simple enough, if you try it manually it may require some effort. For interested readers, you may use this example as an exercise and refer to [3] for a detailed description of the inference process in Bayesian networks.

Let the prior probabilities for B and S be $\{P(B) = 0.8, P(\neg B) = 0.2\}$ and $\{P(S) = 0.1, P(\neg S) = 0.9\}$.

The conditional probabilities for $P(G|SM)$ is given in Table 1.

Similarly we can specify the conditional probabilities for $P(D|G)$ in Table 2.

Table 1 The conditional probability values for $P(G/SM)$

	SM	$\neg SM$
G	(1, 0)	(0.1, 0.9)
$\neg G$	(0, 1)	(0.9, 0.1)

Table 2 The conditional probability values for $P(D|G)$

	G	$\neg G$
D	(1, 0)	(0.1, 0.9)
$\neg D$	(0, 1)	(0.9, 0.1)

Over the last 15 years many inference systems have been developed using Bayesian networks[16]. The applications include diagnostic inference (from effects to causes), causal inference (from causes to effects), intercausal inferences which consider the effect of multiple causes, and mixed inferences which combine the above.

However, work on Bayesian networks still remains primarily academic. In addition to establishing conditional independence based on the d-separate criterion,to use Bayesian networks we must also specify probability distributions which in most cases are very difficult. It has been proven that the exact probabilistic inference using an arbitrary Bayesian network is NP-hard[20]. In fact Dagum and Luby have shown that even approximating conditional probability inference using the Bayesian network is NP-hard and in general intractable[20]. Recently researchers have developed machine learning techniques for constructing Bayesian networks[21,22].

Furthermore, for Bayesian networks there is no mechanism available to handle feedback cycles[18]. 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 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 uncertainties such as vagueness, ambiguity, and granulation. In such a framework, events in a set are considered all equal and assigned the same binary value: yes or no. This, however, bears very little relevance to many real-world problems. For instance, it is in general impossible to model the following problems: The the average income of the community is high, what is the probability that a member of the community lives below the poverty line? If the average income of the community is $50,000, it is highly likely that there is a golf club. How likely there is a golf club in a community with an average income around $45,000? It is safe to say that such problems are ubiquitous in real-world applications.One possible solution to these problems is the fuzzy cognitive maps (FCMs).

Fuzzy Cognitive Maps

In the mid 1980's Kosko[9] introduced the fuzzy cognitive map (FCM) which incorporates fuzzy causality measures in the original cognitive maps. FCM provides a flexible and more realistic representation scheme for dealing with knowledge.FCM provides a mechanism for representing degrees of causality between events/objects.This enables the constructed paths to propagate causality in a more natural fashion through the use of such techniques as forward and backward chaining. In this paper, I briefly introduce an object-oriented FCM for knowledge representation and adaptive inference[1]. The fuzzy cognitive maps are useful for problem domains that can be represented by concepts depicting social events, and causal links from a

[1] Interested readers may refer to [23] for more details.

single partial ordering. In many applications, we will have to deal problems with measurable and non-measurable concepts (variables). In addition such problems may contain many causal descriptions from many causal types and partial orderings.Such a structure is useful to human experts who will need to base their decisions on social,socio-economic and physical concepts, and the causality that exists between these concepts. Fig.5 shows an example of a segment of the FCM we have developed for decision support using GIS data[12,14]

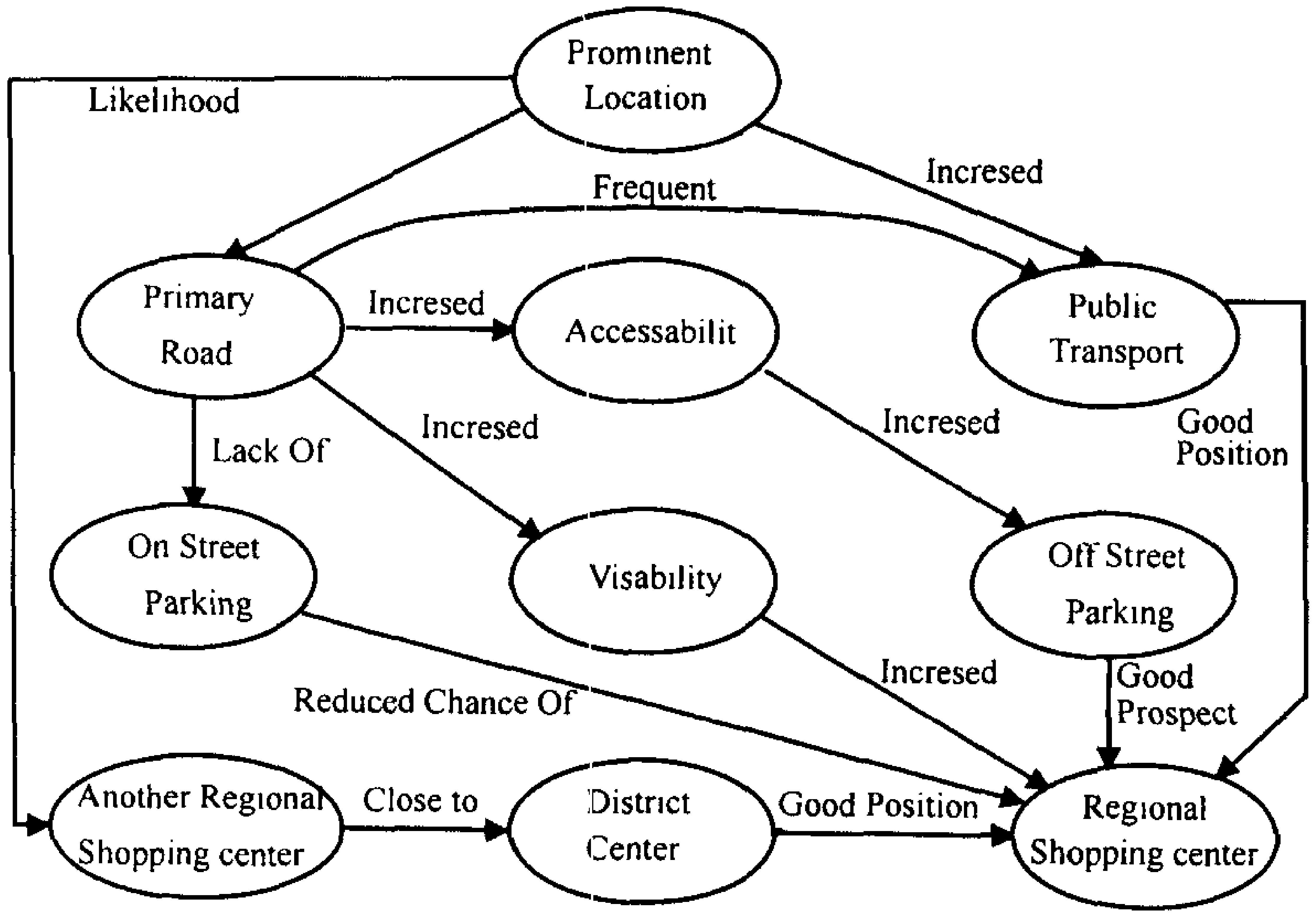


Fig.5 An FCM for modeling shopping center decision

3 State Space in FCMs

In real-world applications, fuzzy cognitive maps are usually very large and complex, containing a large number of concepts and arcs. However, the current method for constructing and analyzing fuzzy cognitive maps are inadequate and infeasible in practice. Furthermore, as the FCM is a typical nonlinear system, the combination of several inputs or initial states may result in new patterns with unexpected behaviors. Systematic and theoretical approaches are required for the analysis and design of fuzzy cognitive maps. In this paper, I analyze the causal inferences in FCMs. A general FCM can be divided into several basic FCMs. The dynamics of a basic FCM is determined by the key vertexes in the FCM. A group of recurrence formulas are given to describe the dynamics of the key vertexes.

The FCM is a digraph in which nodes represent concepts and arcs between the nodes indicate causal relationships between the concepts. The connectivity of the FCM 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 FCM can determine their states. This gives FCMs ability to infer causally. As suggested by Kosko[24,25],this can be determined simply by a threshold. In general, we define a vertex function as follows.

Definition 1. A *vertex function* f_{T,v_i} is defined as

$$x_i = f_{T,v_i}(\mu) = \begin{cases} 1 & \mu > T \\ 0 & T \leqslant \mu \end{cases},$$

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

Usually, if $T = 0$, f_{T,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 FCM can be defined as a weighted digraph with vertex functions. $\mho$ denotes an FCM, $v(\mho)$, or simply, v stands for vertex (or concept) of $\mho$; $V(\mho)$ represents the set containing all the vertices of $\mho$; $x(\mho)$ or x is the state of $v(\mho)$; $\phi = (x_1, \cdots, x_n)^{\mathrm{T}}$ denotes the state of $\mho$, where x_i is the state of vertex v_i and n is the number of vertices of $\mho$; $a(\mho)$ or a stands for an arc in $\mho$; $A(\mho)$ represents the set containing all the arcs of $\mho$; $v^O(a(\mho))$ is the start vertex of $a(\mho)$ and $v^I(a(\mho))$ is the end vertex of $a(\mho)$.

As every vertex may take a value in $\{0, 1\}$, the state space of the FCM is $\{0, 1\}^n$, denoted by $X^0(\mho)$ or X^0. If the state of $\mho$ is ϕ after k inference steps from an initial state ϕ_0, we say that ϕ can be reached from ϕ_0, or $\mho$ can reach ϕ after k inference steps. Although the initial state ϕ_0 of the FCM 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 $(* \ * \ 1 \ 1)^{\mathrm{T}}$ cannot be reached in the FCM shown in Fig.6, where $*$ means that it can be any value in $\{0, 1\}$.

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

$$X^\infty(\mho) \subset X(\mho) \subset X^0(\mho).$$

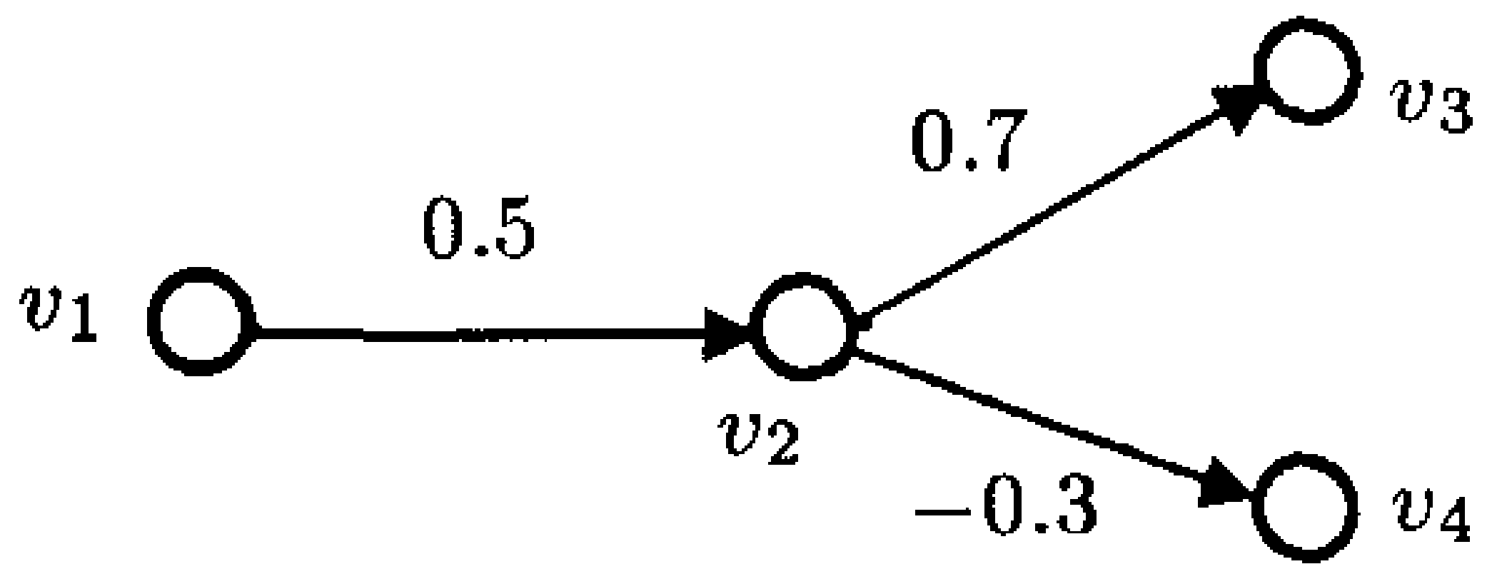


Fig.6 State space of fuzzy cognitive map

It is easy to see which state can be reached by the FCM in Fig.6. However, it will be difficult if the FCM contains a large number of concepts and complex connections. Whether a state can be reached in a given FCM or not is a fundamental and important problem in the analysis and design of FCMs.

Clearly, if a state $\phi^* \in X(\mho)$, there exists a state ϕ_0 such that ϕ^* can be reached in one step inference if ϕ_0 is set as the initial state. The correctness of this assertion can be proved as follows. Since $\phi^* \in X(\mho)$, there exists a state sequence in $X(\mho)$,

$$\phi^*(0), \phi^*(1), \cdots, \phi^*(k) = \phi^*.$$

Obviously, $1 \leqslant k$.

Select a new initial state as $\phi(0) = \phi^*(k-1)$, then the next state $\phi(1)$ will be ϕ^*.

Renumbering the vertices of $\mho$ we can write ϕ^* slightly differently:

$$\overline{\phi} = (\phi_+^{\mathrm{T}}, \phi_-^{\mathrm{T}})^{\mathrm{T}},$$

where $\phi_+ = (1, 1, \cdots, 1)^{\mathrm{T}}, \phi_- = (0, 0, \cdots, 0)^{\mathrm{T}}$. Denote $\overline{W}$ as the adjacency matrix of the renumbered FCM, $n_+ = dim(\phi_+), n_- = dim(\phi_-)$, $\overline{W_+} = (\overline{W}_1^{\mathrm{T}}, \cdots, \overline{W}_{n_+}^{\mathrm{T}})^{\mathrm{T}}, \overline{W_-} = (-\overline{W}_{n_++1}^{\mathrm{T}}, \cdots, -\overline{W}_{n_++n_-}^{\mathrm{T}})^{\mathrm{T}}$. $\phi^* \in X(\mho)$ if and only if

$$\begin{cases} \overline{W}_+ \cdot \psi > 0 \\ \overline{W}_- \cdot \psi \geqslant 0 \end{cases}$$

has a solution ψ in $\{0, 1\}^n$.

From the above discussion, we may develop an algorithm to determine whether a state can be reached from a particular initial state.

4 Causal Module of FCM

A general FCM may contain a large number of vertices with very complicated connections. It is difficult to be handled directly, if at all possible. However, an FCM can be divided into several basic modules, which will be explicitly defined below. Every causal module is a smaller FCM. Vertices (or concepts) of a causal module infer each other and are closely connected. Basic FCM modules are the minimum FCM entities that cannot be divided further.

An FCM $\mho$ is "divided" as FCM $\mho_1$ and FCM $\mho_2$ if

$$\begin{aligned} &1)\quad V(\mho) = V(\mho_1) \cup V(\mho_2), \\ &2)\quad A(\mho) = A(\mho_1) \cup A(\mho_2) \cup B(\mho_1, \mho_2), \end{aligned}$$

where

$$\begin{aligned} B(\mho_1,\mho_2) = \{\ a(\mho)|\ &V^I(a(\mho)) \in V(\mho_1), V^O(a(\mho)) \in V(\mho_2), \\ \text{or}\qquad &V^I(a(\mho)) \in V(\mho_2), V^O(a(\mho)) \in V(\mho_1)\ \}, \\ V(\mho_1) \cap V(\mho_2) = \oslash,&\quad A(\mho_1) \cap A(\mho_2) = \oslash, \\ A(\mho_1) \cap B(\mho_1,\mho_2) = \oslash,&\quad A(\mho_2) \cap B(\mho_1,\mho_2) = \oslash. \end{aligned}$$

This operation is denoted as $\mho = \mho_1 \leftarrow \underset{\oplus}{\underline{\;B(\mho_1,\mho_2)\;}} \rightarrow \mho_2$.

Particularly, we consider the causal relationships in subgraphs:

$$B(\mho_1,\mho_2) = \{\ a(\mho)|V^I(a(\mho)) \in V(\mho_1), V^O(a(\mho)) \in V(\mho_2)\ \},$$

this case is described as "$\mho_2$ is caused by $\mho_1$", or "$\mho_1$ causes $\mho_2$". Such a division is called a *regular division*, denoted as $\mho = \mho_1 \underset{\oplus}{\underline{\;B(\mho_1,\mho_2)\;}} \rightarrow \mho_2$.

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

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

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

From the Definitions 2 and 3, we can see that an FCM containing circles can always be regularly divided into basic FCMs. In general, the basic FCMs of $\mho$ can be ordered concatenately as follows.

$$\mho = \mho_1 \underset{\oplus}{\underline{\;B(\mho_1,\mho_2)\;}} \rightarrow \mho_2\ \underset{\ominus}{\underline{\;B(\mho_2,\mho_3)\;}} \rightarrow \mho_3\ \cdots\ \underset{\oplus}{\underline{\;B(\mho_{m-1},\mho_m)\;}} \rightarrow \mho_m.$$

We establish a formal result in the following theorem.

Theorem 1. *Suppose FCM $\mho$ can be regularly divided into m basic FCMs, then*

$$\mho = (\cup_{i=1}^m \mho_i)\ \cup\ (\cup_{i=2}^m\ \cup_{j=1}^i\ B(\mho_j,\mho_i)), \tag{1}$$

where $\quad V(\mho_i)\ \cap V(\mho_j)\ =\ \oslash\quad i \neq j,\quad A(\mho_i)\ \cap A(\mho_j)\ =\ \oslash\quad i \neq j.$

$$B(\mho_i,\mho_j) = \{a(\mho)|V^I(a(\mho)) \in V(\mho_i), V^O(a(\mho)) \in V(\mho_j)\}.$$

The inference pattern of a basic FCM $\mho_i$ is determined by its input (external) and initial state. The inputs of $\mho_i$ can be determined once the inference pattern of $\mho_k, k < i$ are known. Subsequently, the inference pattern of $\mho_i$ can be analyzed. If we know the inference patterns of the basic FCMs individually, we will be able to obtain the inference pattern of the entire FCM, because the basic FCMs collectively contain all the concepts of the FCM: $V(\mho) = \cup_{i=1}^m V(\mho_i)$.

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

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

Theorem 2 provides not only a rule for determining whether an FCM is a basic FCM or not, but also presents an approach to regularly dividing an FCM if it is not a basic FCM: $In(v_0)$ and $Out(v_0)$. This is illustrated in Fig.8. The FCM in Fig.8(a) can be regularly divided into FCM in Fig.8(b) and in Fig.8 (c), respectively, where Fig.8(b) is $In(v_9)$ and Fig.8 (c) is $Out(v_9)$.

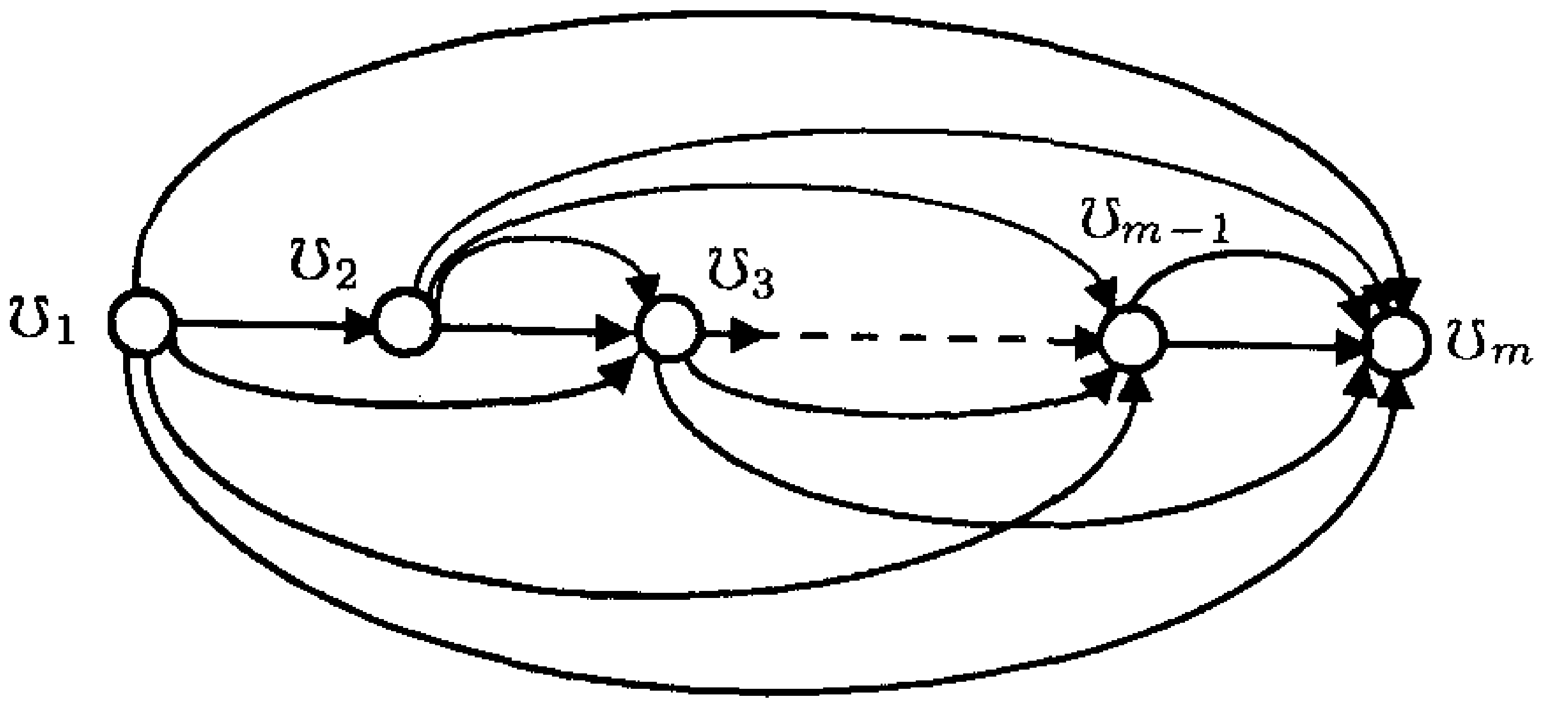


Fig.7 Causal module of FCM

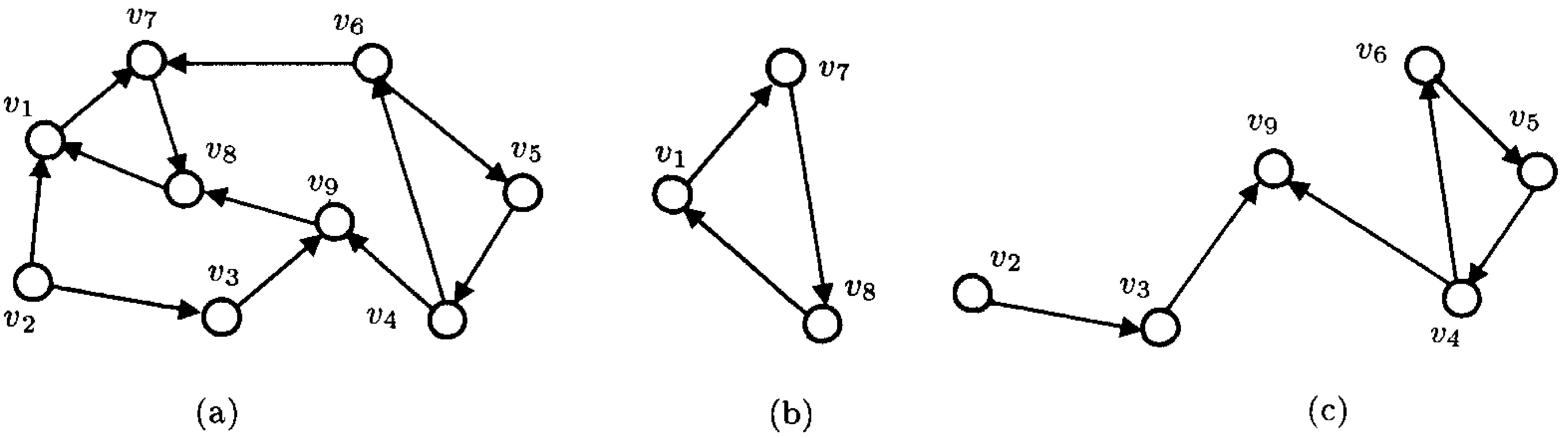


Fig.8 Regularly divided and basic FCMs: according to vertex v_9, FCM in (a) is divided as $Out(v_9)$ in (b) and $In(v_9)$ in (c)

More specifically such a division can be done by the following algorithm.

Algorithm 1

```
Step 0.  If Ʊ is a simple FCM stop.
Step 1.  Select a vertex $v$ of the Ʊ, mark $v$.
Step 2.  Form $In(v)$.
Step 3.  If there is no circle in $In(v)$, go to step 7.
Step 4.  Form $Out(v)$.
Step 5.  If there is no circle in $Out(v)$, go to step 7.
Step 6.  Ʊ is regularly divided into $In(v_9)$ and $Out(v_9)$, stop.
Step 7.  If there is no unmarked vertices, Ʊ is a basic FCM, stop.
Step 8.  Select an unmarked vertex $v$, mark $v$, go to step 2.
```

In general, an FCM can be regularly divided into basic FCMs by repeatedly implementing Algorithm 1.

5 Inference Patterns of Basic FCMs

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

$$\phi(k+1) = f[W\phi(k)] = (f_1[W_1\phi(k)], \cdots, f_n[W_n\phi(k)])^{\mathrm{T}}.$$

However, since in most real applications the FCM 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 FCM is a simple FCM, 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 FCM.*

The Proof of Proposition 1 is obvious. Consequently, the following is 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 FCMs are assumed as basic FCMs, with input paths being standardized and affected branches trimmed. For the sake of simplicity, we assume that all FCMs do not have external input sequences unless they are specifically indicated. FCMs with external input sequences can be analyzed in the similar way.

In our study of the inference pattern of the FCM, 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 FCM 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 sa 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, $v_0 \in V(\mho)$,is not on an input path, then it is on a circle.*

Lemma 2. *Suppose that v_0 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_0 via a normal path, and there is only one normal path from v^* to v_0.*

Again suppose the key-vertex set is $\{v_1, \cdots, v_r\}$. If $v_j (1 \leqslant j \leqslant r)$ 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_i to v_j as $\overline{P^0_{i,j}}, \overline{P^1_{i,j}}, \cdots, \overline{P^{r(i,j)}_{i,j}}$, where $r(i,j)$ is the number of normal paths from v_i to v_j, $\overline{P^0_{i,j}} = P^{<>}_{i,j}$ means that there is no normal path from v_i to v_j, and $f_{\overline{P^0_{i,j}}} = 0$.

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

$$x_j(l) = f_{T_j, v_j}(\sum_{l=1}^{r} \sum_{s=0}^{r(l,j)} f_{\overline{P^s_{l,j}}}\{x_l(l - L(\overline{P^s_{l,j}}))\}. \tag{2}$$

If v_j is the end of an input branch,

$$x_j(l) = f_{T^1_j, T^2_j, v_j}(\sum_{l=1}^{r} \sum_{s=0}^{r(l,j)} f_{\overline{P^s_{l,j}}}\{x_l(k - L(\overline{P^s_{l,j}}))\} + f_{\overline{P_{v(j),j}}}(x_{v(j)}(l - L(\overline{P_{v(j),j}})))). \tag{3}$$

Therefore the inference pattern of the FCM 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 FCM is also determined.

Theorem 4. *Suppose that $K_v(\mho)$ is the key vertex set of $\mho$, $I_v(\mho)$ is the input vertex set of $\mho$, $v_0 \in V(\mho)$, $v_0 \notin K_v(\mho)$ and $v_0 \notin I_v(\mho)$. There exists one and only one vertex x^*, $x^* \in K_v(\mho)$ or $x^* \in I_v(\mho)$ such that there exists a path, $P^*(v^*, v_0)$ from v^* to v_0 via no vertices in $K_v(\mho)$.*

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 FCM is determined. With the help of Theorems 2, 4 and 5, we discuss some important causal inference properties of typical FCMs in the following sections.

6 Inference Pattern of General FCMs

When considering the inference pattern of a general FCM($\mho$), we should first regularly divide it into basic FCMs ($\mho_i, 1 \leqslant i \leqslant m$) and then determine the inference pattern one by one according to the causal relationships between of them. For the basic FCM($\mho_i$), the external input should be formed according to the inference pattern of $\mho_j$ ($1 \leqslant j < i$). Then the input paths should be standardized. After this process, we delete all the affected branches to simplify the FCM($\mho_i$) for further analysis.

From the simplified FCM, we need to construct the key-vertex set. If the basic FCM 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 11. If the basic FCM 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 *ith* column of W has at least two non-zero elements, v_i has at least two input arcs. If $T_{ii} \neq 0$, as the following proposition indicates, v_i is on a circle.

Proposition 3. *Vertex v_i is on a circle if and only if $T_{ii} \neq 0$, where T_{ii} is the ith row and ith column element of matrix $T = \sum_{i=k}^{n} W^k$.*

Following Equations (2) and (3) in Theorem 3, we can obtain the inference pattern of key vertices. Subsequently the states of other vertices including those on the affected branches can be determined accordingly.

The steps for analyzing the inference pattern of an FCM are given below.

Algorithm 2

```
Step 1.  Divide the $\mho$ regularly into basic FCMs: $\mho_1, \cdots, \mho_m$:
             $\mho = (\cup_{i=1}^{m} \mho_i) \cup (\cup_{i=2}^{m} \cup_{j=1}^{i} B(\mho_j, \mho_i))$.
Step 2.  $i = 1$.
Step 3.  Delete the attached branches of $\mho_i$.
Step 4.  Construct new inputs from basic FCMs $\mho_j$ by $B(\mho_j, \mho_i(j < i))$
Step 5.  Standardize the input paths as normal paths.
Step 6.  Construct the key-vertex set.
Step 7.  Determine the state-sequence formula of the key vertices according to Theorem 2.
Step 8.  Determine the state-sequence formula of the remaining vertices.
Step 9.  Determine the state-sequence formula of the affected branch.
Step 10. $i = i + 1$, if $i < m$, go to step 2, else stop.
```

The algorithm is illustrated by the following example. In the example, all the arcs are assumed to be positive, i.e. $w_{i,j} > 0$.

Example

The FCM $\mho_a$ shown in Fig. 9 can be simplified as $\mho_b$ by trimming off the affected branches.

FCM $\mho_c$ and $\mho_d$ in Fig. 10 are the two basic FCMs in $\mho_b$ shown in Fig.9. Fig. 11 shows $\mho_e$ which is $\mho_c$ minus the only affected branch, $AB(v_{20})$. v_{18} is the only key vertex of $\mho_e$:

$$x_{18}(l) = f_{18}(x_{18}(l-4) + x_{18}(l-6)).$$

As 2 is the common factor of 4 and 6, the final state sequence of v_{18} is: $0, 1, 0, 1, \cdots$, $0, 1, 0, 1, \cdots$. The remaining vertex states of $\mho_e$ can be completely determined by x_{18}. For example,

$$x_1(l) = x_{18}(l-2).$$

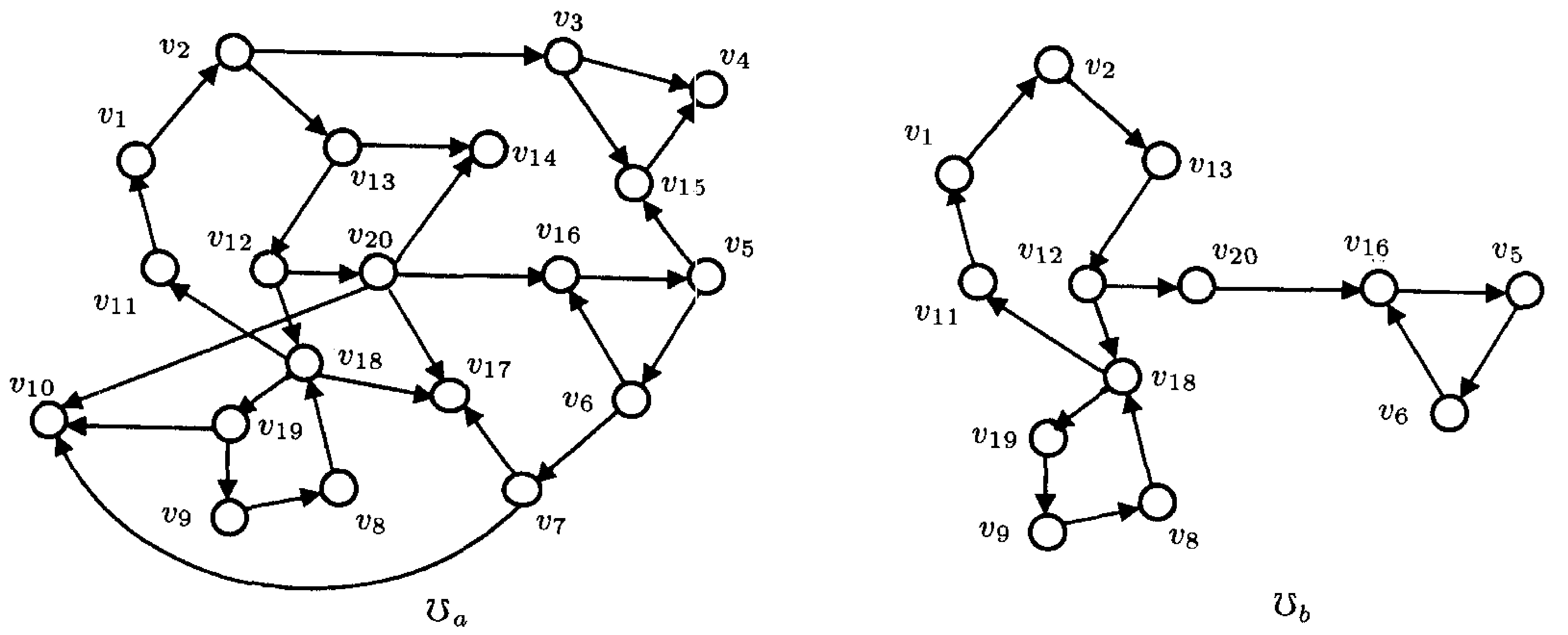


Fig.9 Example of analyzing inference pattern of FCMs: $\mho_a$ is simplified as $\mho_b$ by trimming the affected branches of $\mho_a$

The final state sequence of v_1 is also $0,1,0,1,\cdots,0,1,0,1,\cdots$.

The affected branch $AB(v_{20})$ of $\mho_c$ contains only one vertex: v_{20}. Its state is determined by $\mho_e$, or more specifically, by v_{12}.

$$x_{20}(l) = x_{12}(l-1).$$

The final state sequence of v_{20} is also: $0,1,0,1,\cdots,0,1,0,1,\cdots$.

After all the state patterns of $\mho_c$ are determined, we can reconstruct inputs for basic FCM $\mho_d$. It is shown in Fig.11 as $\mho_f$. The key vertex of $\mho_f$ is v_{16}. By Theorem 2

$$x_{16}(l) = f_{16}(x_{16}(l-3) + x_{20}(l-1)).$$

As the common factor of 2 and 3 is 1, the final state of $\mho_f$ is

$$x_6 \equiv x_{16} \equiv x_5 \equiv 1.$$

With all the state patterns of $\mho_b$ being determined, it is easy to obtain the state pattern for the vertices $(v_3, v_4, v_{15}, v_{14}, v_7, v_{17}, v_{10})$ in the affected branches of $\mho_a$ by the state of $\mho_b$ and it is omitted here.

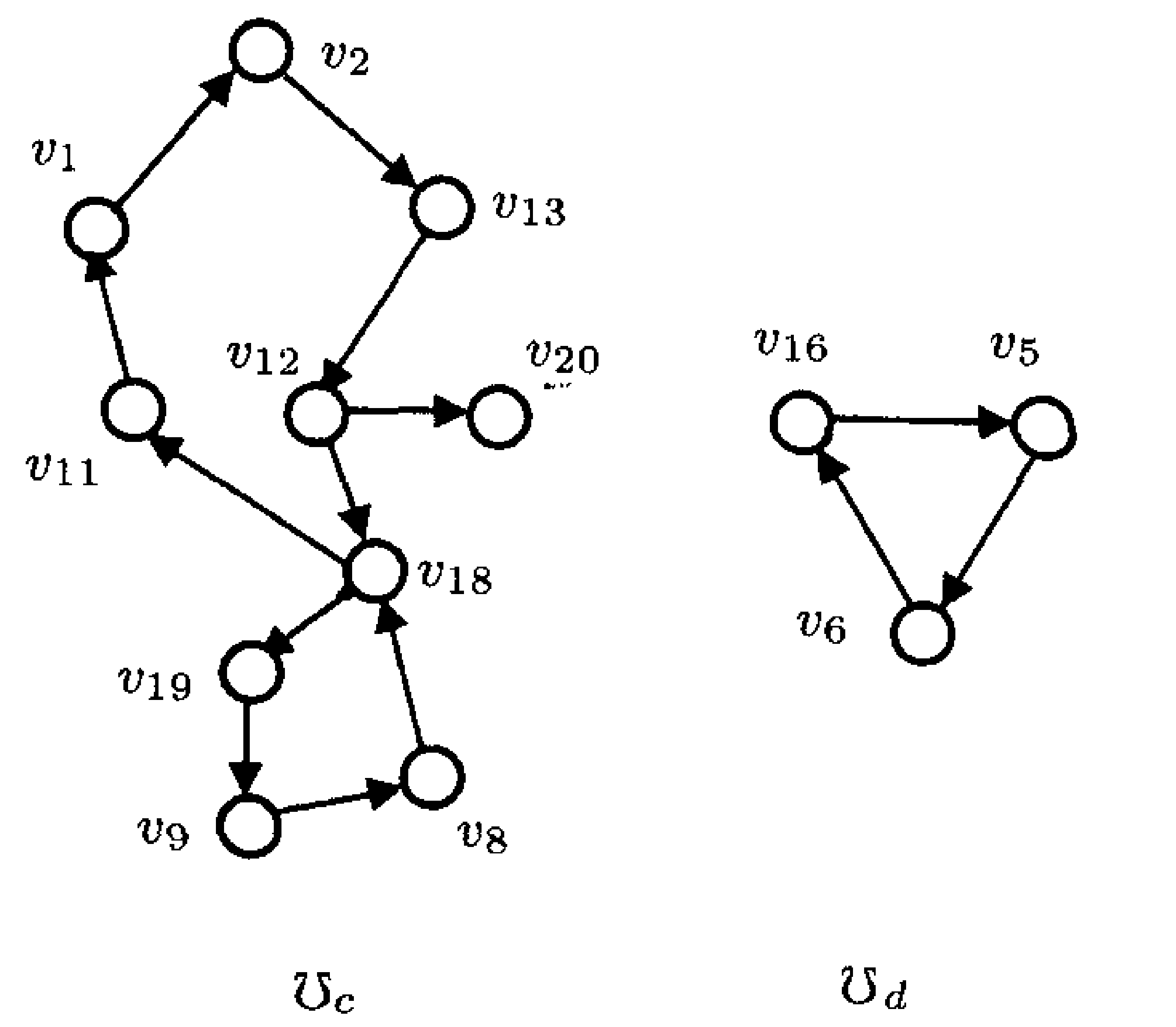


Fig.10 Example of analyzing inference pattern of FCMs: $\mho_b$ can be regularly divided into two basic FCMs: $\mho_c$ and $\mho_d$

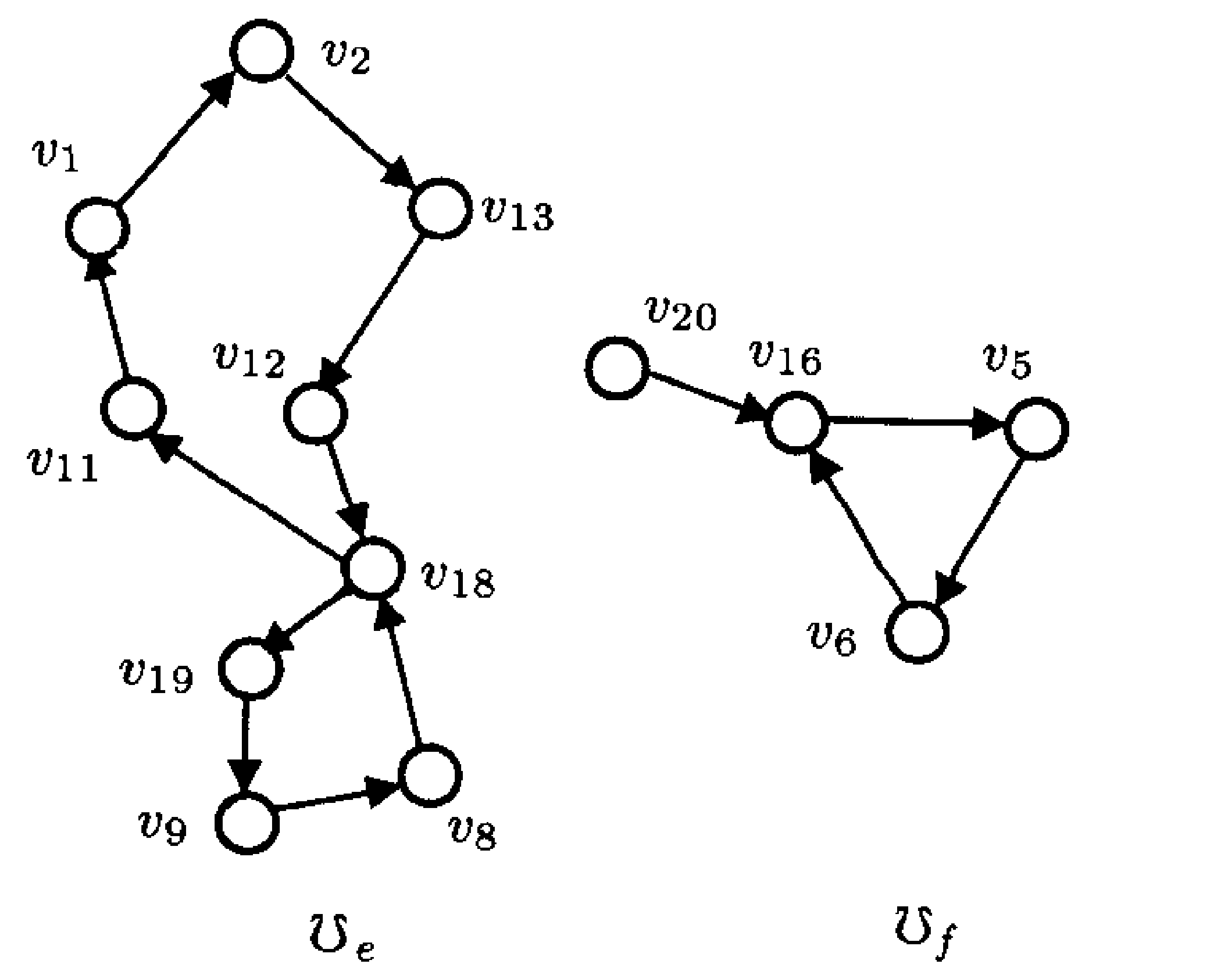


Fig.11 Example of analyzing inference pattern of FCMs:$\mho_e$ is obtained by deleting the affected branch of $\mho_c$ $\mho_f$ is derived from $\mho_d$ by reconstructing the input according to the inference pattern of $\mho_c$

7 Conclusions

In this paper I have given a brief discussion of the theory of causation, and its importance in decision-support and causal discovery systems. One of the well-known causal frameworks is the Bayesian network which uses conditional probability to model causal relationships and uncertainty of the system. Whereas the Bayesian network has received consideration attention in the research community, it still remains an illusive goal for developing robust, functional systems for real-world applications.One of the most significant problems is its inability to model vagueness, ambiguity, and natural descriptions. Fuzzy cognitive maps offer an alternative solution to the problem. I have presented a simple discussion of the causal inference process using FCMs. A general FCM may be very complex, but it can be regularly divided into several basic FCMs according to the results in Section 4. The inference pattern of a basic FCM is the dominant state sequence of some key vertices whose behaviors are described by a general recursive formula. Based these results, I have analyzed the inference patterns of several typical FCMs with an algorithm.

References

1 Pearl J. A contraint-propagation approach to probablistic reasoning. In: Kanal L M & Lemmer J (eds). Uncertainty in Artificial Intelligence. Amsterdam: North-Holland, 1986. 357~370

2 Pearl J. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 1986, **29**(3): 24~288

3 Pearl J. Probablistic Reasoning in Intelligence Systems. Morgan Kaufmann, San Mateo: CA, 1988

4 Andreassen S, Jensen F V, Andersen S K, *et al.* MUNU - an expert EmG assistant. In: Desmedt J E (ed). Computer-aided Electromyography and Expert Systems. 1989. 255~277

5 Rasmussen L K. BOBLO: an expert system based on Bayesian networks to blood group determination of cattle [Research Report No.16]. Tjele. Denmark: Research Center Foulum. 1995

6 Binford T, Levitt T, Mann W. Bayesian inference in model-based machine vision. In: Lemmer J F & Kanal L M (eds). Uncertainty in Artificial Intelligence Amsterdam: Elsevier Science, 1988

7 Levitt T, Hedgcock M, Dye J, *et al.* Bayesian inference for model-based segmentation of computed radiographs of the hand. *Artificial Intelligence.*, 1993, **5**(4): 365~387

8 Heckerman D. Probabilistic Similarity Networks. Cambridge, Massachusetts: MIT Press, 1991

9 Kosko B. Fuzzy cognitive maps. *Int. J. Man-machine Studies*, 1986, **24**: 65~75

10 Axelrod R. Structure of Decision: the Cognitive Maps of Political Elites. Princeton, NJ:Princeton University Press, 1976

11 Kosko B. Adaptive inference in fuzzy knowledge networks. In: Proc. 1st Int. Conf. Neural Networks, 1987, **2**: 261~268

12 Liu Z Q, Satur R. Contextual fuzzy cognitive map for decision support in geographic information systems. *IEEE Trans. Fuzzy Systems*, 1999, **7**(5): 495~507

13 Ndousse T D, Okuda T. Computational intelligence for distributed fault management in networks using fuzzy cognitive maps. In: Proc. IEEE Int. Conf. Communications Converging Technologies for Tomorrow's Application, New York: 1996. 3: 1558~1562

14 Satur R, Liu Z Q. A context-driven intelligent database processing system using object oriented fuzzy cognitive maps. *Int. J. of Intelligent Systems*, 1996, **11**(9): 671~689

15 Satur R, Liu Z Q. A contextual fuzzy cognitive map framefwork for geographic information systems. *IEEE Trans. Fuzzy Systems*, 1999, 7(5): 481~494

16 Russell S, Norvig P. Artificial Intelligence. Prentice-Hall, NJ, 1995

17 Geiger D, Verma T, Pearl J. Identifying independence in Bayesian networks. *Networks*, 1990, **20**(5): 507~534

18 Jensen F V. An Introduction to Bayesian Networks. Springer-Verlag, Berlin, 1996

19 Cooper G F. The computational complexity of probabilistic inference using Bayesian networks. *Articial Intelligence*, 1990, **42**: 393~405

20 Dagum P, Luby M. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Articial Intelligence*, 1990, **60**: 141~153

21 Cooper G F, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 1992, **9**: 309~347

22 Heckerman D, Geiger D, Chickering D M. Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 1995, **20**: 197~243

23 Miao Y, Liu Z Q. On causal inference in fuzzy cognitive map. *IEEE Trans. Fuzzy Systems*, 2000, **8**(1): 107~119

24 Kosko B. Fuzzy system as universal approximators. In: Proc. of the 1st IEEE Int. Conf. Fuzzy Systems, 1992, 1153~1162

25 Kosko B. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall, Englewood Cli, 1992

26 Glymour C, Cooper G. (Eds) Computation, Causation, and Discovery. Cambrige, MA: AAAI Press/MIT Press, 1999

27 Liu Z Q, Miao Y. Fuzzy cognitive map and its causal inference. In: Proc. IEEE Int. Conf. Fuzzy Systems Seoul Korea, 1999, **3**: 1540~1545

28 Pearl J. Causality. Cambrige UK: Cambrige University Press, 2000

因果关系，贝叶斯网络与认知图

刘志强

(墨尔本大学计算机科学与软件工程系智能系统实验室 澳大利亚)

(E-mail: z.liu@computer.org)

摘 要 因果关系在预测和推理中具有重要的作用. 贝叶斯网络已被用于构建诊断和决策系统. 近年来模糊认知图得到了重视. 模糊认知图为结构性知识与因果推理提供了又一个理论框架. 本文简单介绍贝叶斯网络与认知图及其推理方法在智能系统中的应用.

关键词 贝叶斯网络，模糊认知图，因果关系，推理，决策，动态系统，智能系统.

LIU Zhi-Qiang Received M.A.Sc.degree in Aerospace Engineering from the Institute for Aerospace Studies, the University of Toronto, and Ph.D.degree in Electrical Engineering from the University of Alberta, Canada. He was a senior systems engineer in electronics and communications industry in China and Canada. He worked on the following major research and development projects: Inertia-based guidance systems for navigation; Automated manufacturing systems; Channel equalization schemes in wireless and mobile communications; CDMA coding scheme initiatives (written in *C* programming language), which was supported in part by the Canadian Government; Wireless data communications, in particular, for mobile office scenarios; and Internet/Intranet wireless data communications in enterprises. He is currently on leave as a Professor with School of Creative Media, City University of Hong Kong. He is a senior member of IEEE and has served over a dozen major international conferences as Chairman or committee member. He is now on several editorial boards of international journals. He has received a number of prestigious fellowships, scholarships, and an award from the International Pattern Recognition Society for an outstanding contribution. His research interests include machine intelligence, fuzzy-neural systems, visual communications, wireless networks and applications, web/data-mining.