

An Information Theoretic Approach to Verification of Modular Bayesian Fusion Systems

Patrick de Oude

Intelligent Autonomous Systems Group, Informatics Institute
University of Amsterdam, The Netherlands
Email: poude@science.uva.nl

Gregor Pavlin

Thales Research & Technology Netherlands
Delftechpark 24, 2628 XH, Delft, The Netherlands
Email: gregor.pavlin@icis.decis.nl

Abstract—This paper introduces an information theoretic approach to verification of causal models in modular Bayesian fusion systems. We assume distributed fusion systems which are gradually extended by adding new modules, each having a limited domain knowledge captured in local Bayesian networks. However, since different modules originate from different, independent design processes important dependencies between the variables in different modules might not correctly be captured in the distributed fusion system. This could have a significant impact on the fusion quality. The introduced method supports discovery of significant dependencies which are ignored in the distributed fusion system.

Keywords: (distributed) Bayesian networks, mutual information, structure learning.

I. INTRODUCTION

This paper introduces an information theoretic approach to verification of causal models in distributed Bayesian fusion systems. In particular we assume a recently introduced Distributed Perception Networks framework (DPN) [1], [2], a MAS approach to modular Bayesian fusion. DPN agents are basic building blocks which cooperate to form arbitrarily large distributed fusion systems. DPN systems are in essence distributed self organizing Bayesian classifiers, which consist of agents with very heterogeneous domain expertise encoded in local Bayesian networks [3]. DPN fusion agents are used as assistants to human decision makers; they observe the environment, distill the relevant information and supply it to the decision makers.

Such systems are relevant for an increasingly relevant class of situation assessment problems, such as detection of toxic gases, disease outbreaks, fires, etc. In such settings, critical hidden events must be inferred through interpretation (i.e. fusion) of large quantities of uncertain and very heterogeneous information. The information can be accessed via static sensors or ad-hoc sensor networks formed at runtime as sensors are delivered to the area of interest via mobile platforms (e.g. unmanned aerial vehicles). In addition, we might be able to obtain valuable information from humans by using the existing communication infrastructure, such as mobile phone networks, WWW, etc.

Interpretation of different types of information requires adequate domain models which provide mapping

between heterogeneous observations and hypotheses of interest. However, situation assessment in crisis management settings introduces several substantial challenges. Information sources are heterogeneous and noisy. The heterogeneity of observations implies complex domain models. On the other hand, the domain complexity means that models are inevitably abstractions associated with significant uncertainties. In order to be able to achieve reliable detection in such settings, huge quantities of noisy information should be processed. Also, models must adapt to changing constellations of information sources which are often not known prior to the operation. Given these challenges, a modular solution such as the DPN framework seems to be an appropriate choice; fusion modules with limited domain knowledge can assemble adequate domain models at runtime and the processing load can be distributed throughout a system of networked devices.

A. Construction of adequate domain models

One of the major challenges, however, is creation of adequate local domain models which support correct fusion in a distributed system. It has been shown that by considering Markov boundaries we can derive simple rules for partitioning of monolithic Bayesian networks such that globally coherent fusion in distributed systems based on local inference processes is achieved [2]. The design and assembly rules guarantee that whenever a newly supplied module joins a DPN organization, the extended fusion system supports correct inference through asynchronous peer-to-peer communication, without any centralized fusion control.

However, often it is impractical or impossible to first obtain a monolithic domain model and then decompose it into local models; construction of complex monolithic models can require contributions from many different experts and machine learning processes, which can result in inefficient development of fusion systems. It turns out that in certain domains it is possible to create a correct complex distributed model out of basic building blocks in a bottom-up manner, without knowing the corresponding true monolithic model; i.e. we generate adequate local models, which were not obtained through

decomposition. This seems to be possible in case of monitoring processes involving many independent sensory devices. While a model of a particular sensor can be a very complex BN, the components of one sensing device are not influenced by components of another sensing device, etc. In such settings local models pairwise share a few variables. Thus, the designers must merely coordinate the labeling of only a few variables. Despite of this, it is likely that certain dependencies are overlooked. While a local model from a designer perspective might be a perfect model over a subset of variables, it might not support correct fusion in an assembled system due to an omission of an important link between the variables involved in different modules. While sound engineering approaches can reduce the danger of such faults, they cannot prevent them.

B. Discovering dependencies

In this paper we present a method for efficient verification of dependencies in modularized Bayesian fusion models that is based on an information theoretic approach to dependency testing. The presented approach makes use of conditional independence tests in a similar manner as do common structure learning approaches [4], [5]. In general, these approaches are used to learn dependencies between a set of variables from scratch. Unfortunately, these approaches are limited to rather small models due to computational complexity. While the presented method is derived from the *PC-algorithm* [4], it has been adapted to a different class of problems. Namely, the presented method is geared toward detection of inadequately represented or missing dependencies between variables in different fusion modules which are gradually integrated into a distributed fusion system. We assume that local BNs in each fusion module correctly capture all relations between the local variables. By considering the theory on *I-maps* [3], we can achieve efficient detection of inadequately represented dependencies between variables in different modules. In particular, we achieve efficient verification by exploiting the composition/decomposition relations which are valid in case of faithful probability distributions [3].

We provide experimental results based on a simplified gas detection process. We illustrate the impact of missing dependencies on the fusion quality and demonstrate the effectiveness of the presented approach to detection and discovery of missing dependencies in modular Bayesian fusion systems. The presented method exploits mutual information in the sampled data, which requires sufficient quantities of data. Issues regarding the complexity and sampling of sufficient amounts of data are discussed and we illustrate how the presented method could be used in real world settings.

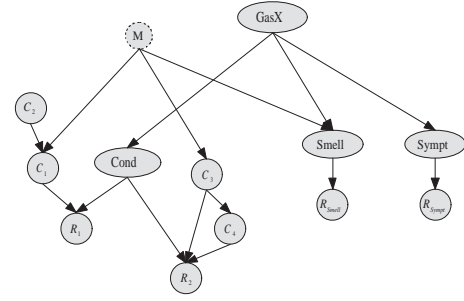


Figure 1. Simplified true causal model for the detection of gas.

II. CAUSAL FUSION MODELS

Many real world fusion problems can often be modeled through causal probabilistic models which explicitly capture stochastic causal processes. For example, consider the gas detection fusion model given in Figure 1 where we want to detect the presence or absence of some toxic gas represented by *GasX*. The existence of *GasX*¹ is represented by binary variable *GasX*, where the instantiations *GasX* = *true* and *GasX* = *false* correspond to the presence and the absence of *GasX*, respectively. In addition, we assume that the system uses two types of chemical sensors. Both types of chemical sensors measure the conductivity (i.e. electrical current) in a semiconductor exposed to the contaminated air; *GasX* reacts with the semiconductor which influences the conductivity. But each sensor type evaluates the signal in a different way. States of the binary variable *Cond* correspond to the situations where electrical current under ideal circumstances would either exceed some detection threshold (i.e. *Cond* = *true*) or be below that threshold (i.e. *Cond* = *false*). A certain level of conductivity *Cond* in the area and the states of the sensor components C_i might cause with certain probability sensors to produce sensor reports, e.g. R_1 and R_2 . Where R_i = *true* asserts an affirmative report about the presence of *GasX*, while R_i = *false* asserts a negative report about the absence of *GasX*.

Besides the chemical sensors we assume that there are humans in the area, who have olfactory reactions to *GasX*. The states of binary variable *Smell* represent situations in which people familiar with a typical smell of *GasX* are either able or unable to recognize the smell under ideal circumstances. In addition, humans might experience certain symptoms *Sympt* caused by *GasX*, such as nausea. Humans might report about the smell or particular symptoms represented by R_{Smell} and R_{Sympt} , respectively. The sensor components and smell, i.e. C_1 , C_3 and *Smell*, respectively, are influenced by the humidity in the air represented by variable *M*. However, we can

¹For the sake of simplicity we say that a toxic gas is present if its concentration exceeds some critical value.

safely assume that the sensing processes do not influence the humidity.

In order to be able to cope with inherent uncertainties, we use (causal) Bayesian networks, probabilistic models which can capture stochastic causal processes in a rigorous and compact way [3]. Note that the presented example is a severe simplification of real world processes in order to facilitate discussion in this paper. On the other hand it is rich enough to illustrate the main concepts.

A. Bayesian networks

A Bayesian network (BN) is an efficient graphical representation of a joint probability distribution (JPD) $P(\mathbf{V})$ over a set of random variables defined in \mathbf{V} which can be used for exact inference [3], [6], [7]. A BN is represented by a tuple (G, \mathbf{P}) , where $G = (\mathbf{V}, \mathbf{E})$ is a Directed Acyclic Graph (DAG) with nodes \mathbf{V} and directed edges $E = (V_i, V_j)$ between nodes in \mathbf{V} , where $E \in \mathbf{E}$. \mathbf{P} is the set of conditional probabilities $P(V_i | \mathbf{Pa}(V_i))$ for all $V_i \in \mathbf{V}$, where $\mathbf{Pa}(V_i)$ represents the parent nodes of V_i in DAG G . Conditional probabilities are often represented through conditional probability tables (CPTs). The joint probability distribution $P(\mathbf{V})$ can be computed through the chain rule for BNs [6]: $P(\mathbf{V}) = \prod_i P(V_i | \mathbf{Pa}(V_i))$.

BNs explicitly capture *conditional independence* [8] between random variables. A variable X is conditionally independent of variable Y given variable Z when $P(X|Y, Z) = P(X|Z)$. Conditional independencies are represented through *d-separation* [3], [6], [7] relations in DAG G :

Definition 1 (D-Separation): If \mathbf{X}, \mathbf{Y} and \mathbf{Z} are three mutually exclusive subsets of nodes from a DAG G , then \mathbf{Z} **d-separates** \mathbf{X} from \mathbf{Y} , denoted by $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_G$, if along every path between a node in \mathbf{X} and a node in \mathbf{Y} there is a node V_i satisfying one of the two conditions:

- 1) V_i has converging arrows and neither V_i nor its descendants are in \mathbf{Z} or,
- 2) V_i does not have converging arrows and $V_i \in \mathbf{Z}$.

The representational explicitness of conditional independence in BNs means that every represented d-separation in a DAG G should have a *valid* corresponding conditional independence relation in $P(\mathbf{V})$. Consequently, the following implication should hold:

$$(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_G \Rightarrow (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_P \quad (1)$$

Note that, the implication in (1) does allow us to model direct dependencies² between nodes in G which are not true in $P(\mathbf{V})$. Thus, the model can contain superfluous dependencies, which in turn means that the parameter estimation and inference can be less efficient. However, in such a case we still capture all dependencies in $P(\mathbf{V})$ and given sufficient amounts of data the true distribution can be correctly encoded. Whenever the implication in

²Note that, two variables that are directly dependent cannot be made conditionally independent given any conditioning set of variables.

(1) for DAG G holds we say that G is an *I-map* of $P(\mathbf{V})$ [3].

Ideally, no superfluous dependencies are captured by the model. In other words, every conditional independence in P corresponds to a certain d-separation in G and vice versa, which can be formulated as follows:

$$(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_G \Leftrightarrow (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_P \quad (2)$$

Whenever Equation (2) holds we say that $P(\mathbf{V})$ is *faithful* [4] to DAG G (or G is a perfect map [3] of P). If a probability distribution is faithful then there exist a DAG G for it that faithfully represents all (conditional) dependencies and independences between the variables in this probability distribution. In this paper we only assume faithful probability distributions. Parameters in a BN (i.e. conditional probabilities) can be estimated with the help of the maximum likelihood principle by considering relative frequencies in some data set \mathcal{D} of m cases. A single case $d_i \in \mathcal{D}$, $i \leq m$ is represented as a configuration of states $\mathbf{v} = (x, y, \dots, z)$ for the domain variables $\mathbf{V} = \{X, Y, \dots, Z\}$.

In this paper we assume that the data set cases $d_i = \mathbf{v}$ are sampled from a generative probability distribution $P(\mathbf{V})$ which can faithfully be represented by a BN with DAG G .

B. Testing conditional independence between variables

Let's assume that a BN models a distribution $P(\mathbf{V})$ from which data set \mathcal{D} was sampled. Moreover, we can use the conditional mutual information (CMI) measure between mutually exclusive sets of variables $\mathbf{X} \subset \mathbf{V}$, $\mathbf{Y} \subset \mathbf{V}$ and $\mathbf{Z} \subset \mathbf{V}$ to test conditional independence $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_P$:

$$I(\mathbf{X}, \mathbf{Y} | \mathbf{Z}) = \sum_{x, y, z} P(x, y, z) \log \frac{P(x, y | z)}{P(x | z)P(y | z)} \quad (3)$$

Note that the probabilities $P(x, y | z)$, $P(x | z)$ and $P(y | z)$ in Equation 3 are estimated using data samples in \mathcal{D} .

By considering (2) we know that a BN correctly captures the true distribution $P(\mathbf{V})$ iff for every $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_G$ the mutual information $I(\mathbf{X}, \mathbf{Y} | \mathbf{Z}) = 0$ and vice versa. *In other words, by using the CMI measure we can use data samples from \mathcal{D} to test I-mapness of causal probabilistic models.*

III. MODULAR FUSION

The gas monitoring process model given in Figure 1 can be represented by a system of collaborating fusion modules based on smaller causal models (i.e. BNs) shown in Figure 2. These fusion models can be constructed by using the design rules described in [2]. Such a system of fusion modules forms a *modular Bayesian network*. The fusion modules (also BN modules) in Figure 2 can be defined as follows:

Definition 2 (BN Module): A **BN module** $\psi_i = (G_i, \mathbf{P}_i)$ is a Bayesian network with DAG $G_i = (\mathbf{V}_i, \mathbf{E}_i)$, where \mathbf{V}_i

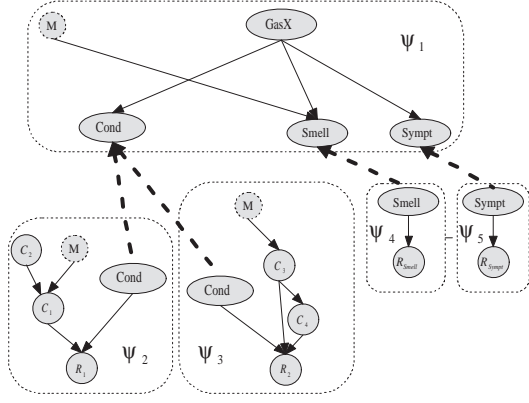


Figure 2. Modularization of the gas detection fusion model in Figure 1

are the variables in G_i and E_i is a set of directed edges $E = (X, Y)$, where $E \in E_i$, $X \in \mathbf{V}_i$ and $Y \in \mathbf{V}_i$. \mathbf{P}_i is a set of (conditional) probability distributions defined for each variable in \mathbf{V}_i .

A BN module encodes probabilistic knowledge over a subset of variables $\mathbf{V}_i \subset \mathbf{U}$, where \mathbf{U} represents all variables in the domain under investigation. Therefore, a BN module is a subgraph of the monolithic BN. The monolithic BN encodes probabilistic knowledge of a JPD $P(\mathbf{U})$, while a BN module encodes probabilistic knowledge of the JPD $P(\mathbf{V}_i) = \sum_{\mathbf{U} \setminus \mathbf{V}_i} P(\mathbf{U})$. For example, in Figure 2 the BN module $\psi_1 = (G_1 = (\mathbf{V}_1, \mathbf{E}_1), \mathbf{P}_1)$ represents the dependencies between the variables in $\mathbf{V}_1 = \{GasX, M, Cond, Smell, Sympt\}$ using graph G_1 with edges in \mathbf{E}_1 . \mathbf{P}_1 represents the (conditional) probabilities $P(GasX)$, $P(M)$, $P(Cond|GasX)$, $P(Smell|GasX, M)$, $P(Sympt|GasX)$.

In this paper we assume that every BN module is a *local I-map* over the variables \mathbf{V}_i :

Definition 3 (Local I-mapness): A BN module $\psi_i = (G_i, \mathbf{P}_i)$ where DAG $G_i = (\mathbf{V}_i, \mathbf{E}_i)$ is a **local I-map** of $P(\mathbf{V}_i) = \sum_{\mathbf{U} \setminus \mathbf{V}_i} P(\mathbf{U})$, where $\mathbf{V}_i \subset \mathbf{U}$ and \mathbf{U} contains all the variables in the domain, if all d-separation relations between the variables in \mathbf{V}_i correspond to valid conditional independencies in $P(\mathbf{V}_i)$.

For example, ψ_1 with variables \mathbf{V}_1 has the local I-map property, because all the represented d-separations, such as $(Cond \perp\!\!\!\perp Smell|GasX)_{G_1}$, $(\{GasX, Sympt\} \perp\!\!\!\perp M|\emptyset)_{G_1}$, $(Cond \perp\!\!\!\perp Sympt|\{GasX, M, Smell\})_{G_1}$ etc. correspond to valid conditional independencies in $P(\mathbf{V}_1)$ assuming that the gas detection fusion model in Figure 1 is the ground truth model. From this point on we assume that the BN in Figure 1 is the ground truth. Collectively, the BN modules must form a *modular Bayesian network*:

Definition 4 (Modular Bayesian networks): A **Modular Bayesian network** Ω is defined as a tuple $(\mathcal{M}, \mathcal{R})$, where \mathcal{M} is the set of BN modules defined in Ω . \mathcal{R} is a finite set of BN module pairs $\{\langle \psi_i, \psi_j \rangle | i \geq 1, j \geq 1, i \neq j\}$ with $\psi_i = ((\mathbf{V}_i, \mathbf{E}_i), \mathbf{P}_i) \in \mathcal{M}$, $\psi_j = ((\mathbf{V}_j, \mathbf{E}_j), \mathbf{P}_j) \in \mathcal{M}$ and $\mathbf{V}_i \cap \mathbf{V}_j \neq \emptyset$. Every module pair $\langle \psi_i, \psi_j \rangle$ represents a link

between the two BN modules ψ_i and ψ_j over which partial fusion results can be communicated. Every modular BN has the following properties:

- each BN module pair is *graph consistent* [7]; symmetric, (i.e. $\langle \psi_i, \psi_j \rangle = \langle \psi_j, \psi_i \rangle$); irreflexive, i.e. the same BN module can not be in a pair (i.e. $\langle \psi_i, \psi_i \rangle$);
- a modular BN cannot contain a loop between BN modules. For example, module pairs $\langle \psi_i, \psi_j \rangle$, $\langle \psi_j, \psi_k \rangle$ and $\langle \psi_i, \psi_k \rangle$ cannot occur simultaneously in \mathcal{R} ;
- every BN module in \mathcal{M} must at least share one variable with another BN module defined in \mathcal{M} ;
- the modular BN has the *running intersection property* (see [6], [7] for a definition).

For example, in Figure 2 the monolithic gas detection fusion model (see Figure 1) is partitioned into five different BN modules: ψ_1 , ψ_2 , ψ_3 , ψ_4 and ψ_5 . The modular BN is then defined as $\mathcal{M} = \{\psi_1, \psi_2, \psi_3, \psi_4, \psi_5\}$ and $\mathcal{R} = \{\langle \psi_1, \psi_2 \rangle, \langle \psi_1, \psi_3 \rangle, \langle \psi_1, \psi_4 \rangle, \langle \psi_1, \psi_5 \rangle\}$.

Reasoning in the modular BN in Figure 2 is equivalent to reasoning in the monolithic BN in Figure 1. I.e. we can map the structure of the modular BN Ω to its corresponding monolithic graph version, which we call G^Ω (for readability we will write G instead of G^Ω in the rest of the paper). The graph in Figure 1 is the G^Ω of the modular BN in Figure 2.

A. Constructing modular Bayesian networks

We assume that the construction of a modular BN $\Omega = (\mathcal{M}, \mathcal{R})$ is accomplished through a certain integration order of BN modules; Ω is extended with a BN module in each extension step:

Definition 5 (Extension Step): Extending a modular BN $\Omega = (\mathcal{M}, \mathcal{R})$ with BN module ψ_i is defined as: $\Omega^* \leftarrow \Omega \sqcup \psi_i$, where ψ_i is connected to a single BN module $\psi_j \in \mathcal{M}$ such that $\Omega^* = (\mathcal{M} \cup \psi_i, \mathcal{R} \cup \langle \psi_j, \psi_i \rangle)$. If $\mathcal{M} = \emptyset$ then $\Omega^* = (\{\psi_i\}, \mathcal{R} = \emptyset)$.

Extending BN modules to modular BN Ω might at some point violate the I-map property in the corresponding monolithic graph G of Ω , i.e. the newly represented d-separation relations in G do not correspond to dependencies in the data. Consider the modularization of the gas detection model in Figure 3. In this modular BN the direct dependence between humidity M and the sensor component C_1 is missing (denoted by dashed directed edge). The BN modules in Figure 3 are all local I-maps, but when connected together the I-map property does not hold anymore. For example, say we want to extend $\Omega = (\mathcal{M} = \{\psi_1\}, \mathcal{R} = \emptyset)$ with the BN module ψ_2 shown in Figure 4(a). In this case, ψ_2 is connected to the already integrated $\psi_1 \in \mathcal{M}$ resulting in $\Omega^* = (\mathcal{M} = \{\psi_1, \psi_2\}, \mathcal{R} = \langle \psi_2, \psi_1 \rangle)$. The corresponding monolithic graph G^* in Figure 4(b) is not an I-map anymore, because it encodes the d-separations $(M \perp\!\!\!\perp C_1|Cond)_{G^*}$, $(M \perp\!\!\!\perp R_1|Cond)_{G^*}$, $(Smell \perp\!\!\!\perp C_1|Cond)_{G^*}$ and $(Smell \perp\!\!\!\perp R_1|Cond)_{G^*}$. These d-separations are not valid in the ground truth model shown in Figure 1.

Consequently, the implication in (1) does not hold and graph G^* is not an I-map. Thus, the modular BN Ω^* does not support correct reasoning. Namely, the dependence between M and C_1 is not captured by the BN modules in Figure 3 correctly.

Important Observation: *given that all BN modules have the local I-map property does not, in general, imply that the modular BN constructed out of these BN modules is an I-map.*

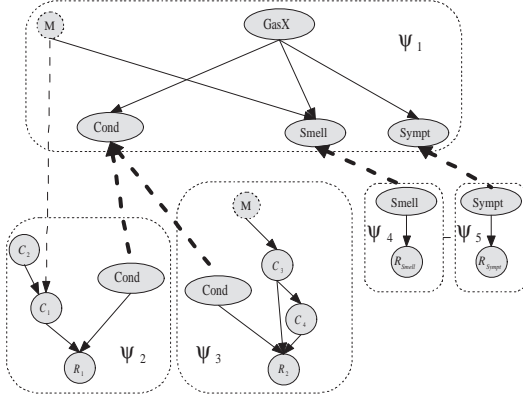


Figure 3. An incorrect modularization of the gas detection fusion model in Figure 1

IV. VERIFICATION OF MODULAR FUSION MODELS

The CMI measure in Section II-B can be used to test I-mapness of a modular BN; i.e. we verify whether a modular BN corresponds to a monolithic BN which is an I-map of the variables represented by this modular BN. We facilitate further analysis by introducing the concept of *pairwise Conditional Independence* (pairwise CI) between the variables from different modules in a modular BN.

Definition 6 (Pairwise CI): Let's assume a modular BN Ω with a set of modules \mathcal{M} defined over variables \mathbf{V} which is extended by adding a new module ψ_i with variables \mathbf{V}_i . In addition, Ω contains module $\psi_j \in \mathcal{M}$ with variables \mathbf{V}_j , such that $\mathcal{S} = \mathbf{V}_i \cap \mathbf{V}_j \neq \emptyset$. For any pair of variables $X \in \{\mathbf{V}_i \setminus \mathcal{S}\}$ and $Y \in \{\mathbf{V}_j \setminus \mathcal{S}\}$ a **pairwise conditional independence (CI)** is defined as:

$$(X \perp\!\!\!\perp Y | \mathcal{S})_P \quad (4)$$

Note that expression (4) describes conditional independence in the true distribution which the modular BN should capture. This conditional independence can be tested with the help of the CMI measure. We illustrate pairwise CI with the help of the modular BN in Figure 2. Let's assume that we wished to extend modular BN $\Omega = (\{\psi_1, \psi_3, \psi_4, \psi_5\}, \{\langle \psi_1, \psi_3 \rangle, \langle \psi_1, \psi_4 \rangle, \langle \psi_1, \psi_5 \rangle\})$ with BN module ψ_2 . In this case we could identify intersection set $\mathcal{S} = \mathbf{V}_1 \cap \mathbf{V}_2 = \{Cond, M\}$. Given that the extended modular BN corresponded to the ground truth model shown in Figure 1, we could detect several pairwise CIs

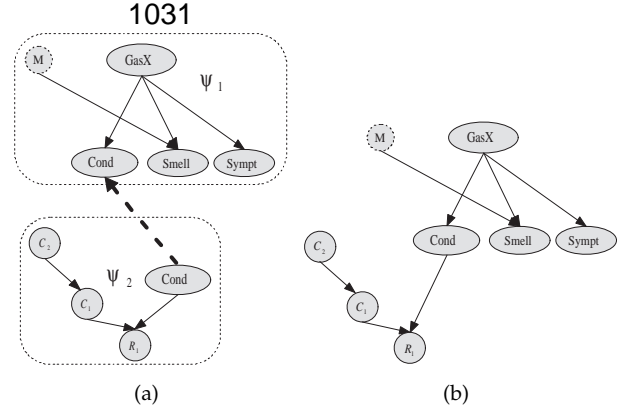


Figure 4. Extending $\Omega = (\mathcal{M} = \{\psi_1\}, \mathcal{R} = \emptyset)$ with BN module (a) and the resulting monolithic graph G^* of $\Omega^* = (\mathcal{M} = \{\psi_1, \psi_2\}, \mathcal{R} = \langle \psi_2, \psi_1 \rangle)$ (b).

between variables from different modules, such as for example $(C_1 \perp\!\!\!\perp R_{Smell} | \mathcal{S})_P$, $(R_1 \perp\!\!\!\perp R_{Smell} | \mathcal{S})_P$, $(C_1 \perp\!\!\!\perp Smell | \mathcal{S})_P$, etc.

Moreover, it turns out that Pairwise CI tests can be used for efficient verification of I-mapness of modular BNs.

Proposition 1 (Pairwise CIs & I-mapness): Assume a modular BN $\Omega = (\mathcal{M}, \mathcal{R})$ with the corresponding I-map $G = (\mathbf{V}, \mathbf{E})$ and a BN module ψ_i with the local I-map $G_i = (\mathbf{V}_i, \mathbf{E}_i)$. Also assume that the JPD $P(\mathbf{V} \cup \mathbf{V}_i)$ is faithful. By extending Ω with ψ_i we get the modular BN Ω^* , i.e. $\Omega^* \leftarrow \Omega \sqcup \psi_i$ where Ω^* corresponds to the monolithic graph G^* . G^* is an I-map if and only if all pairwise CIs (see Definition 6) of the sets $\{\mathbf{V} \setminus \mathcal{S}\}$ and $\{\mathbf{V}_i \setminus \mathcal{S}\}$ are valid, where $\mathcal{S} = \mathbf{V} \cap \mathbf{V}_i$.

Proof: In general, we know that if some distribution $P(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4)$ is faithful the decomposition/composition relations [3] hold: $(\mathbf{X}_1 \perp\!\!\!\perp \mathbf{X}_2 | \mathbf{X}_4)_P \wedge (\mathbf{X}_1 \perp\!\!\!\perp \mathbf{X}_3 | \mathbf{X}_4)_P \Leftrightarrow (\mathbf{X}_1 \perp\!\!\!\perp \{\mathbf{X}_2, \mathbf{X}_3\} | \mathbf{X}_4)_P$. Moreover, let's assume set $\mathbf{X} \subseteq \{\mathbf{V}_i \setminus \mathcal{S}\}$, i.e. a subset of variables from BN module ψ_i and set $\mathbf{Y} \subseteq \{\mathbf{V} \setminus \mathcal{S}\}$ which is a subset of variables from the modular BN Ω , where $\mathcal{S} = \mathbf{V} \cap \mathbf{V}_i$. According to the decomposition/composition relations we know that if every pair $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$ is *pairwise CI* (Definition 6) then we know that any of the sets \mathbf{X} and \mathbf{Y} are conditionally independent given \mathcal{S} as well. Moreover, since the graphs G_i and G are (local) I-maps we know that the new modular BN Ω^* with the set of variables $\mathbf{V} \cup \mathbf{V}_i$ corresponds to a graph G^* which must also be an I-map. ■

In other words, verification based on pairwise CIs is sufficient for determining whether an extended modular BN corresponds to an I-map.

V. DISCOVERY OF MISSING INTER MODULE DEPENDENCIES

Verification of I-mapness of modular BNs can be used as a basis for discovery of modeling faults. Whenever one of these pairwise CIs fail then we know that the modular BN is incorrect and, consequently, certain dependencies between BN modules are missing. In fact,

with the help of pairwise CI tests we can discover all pairs of variables whose dependency is not correctly captured in the modular BN.

Proposition 2 (Edge Discovery): Assume that we extend $\Omega = (M, \mathcal{R})$ corresponding to I-map $G = (V, E)$ with module ψ_i corresponding to the local I-map $G_i = (V_i, E_i)$ to form the new modular BN Ω^* corresponding to graph G^* , i.e. $\Omega^* \leftarrow \Omega \sqcup \psi_i$. Moreover, the pairwise CIs for the sets V of Ω and V_i of ψ_i are tested using the conditioning set $S = V_i \cap V \neq \emptyset$. We test each possible pair of variables. For every test failure between a pair of tested variables $X \in V_i \setminus S$ and $Y \in V \setminus S$ we add the pair (X, Y) to a set of pairs \mathcal{E} . The set \mathcal{E} is guaranteed to contain all the pairs of variables which are directly dependent in the true distribution $P(V \cup V_i)$ but this dependence is not correctly captured by the corresponding modular BN.

Proof: The proof is trivial. Namely, no conditioning set S can make two variables which are directly dependent according to $P(V \cup V_i)$ conditionally independent. Therefore, set \mathcal{E} must contain all edges between variables $X \in V_i \setminus S$ and $Y \in V \setminus S$ which are directly dependent in $P(V \cup V_i)$. ■

Pairs of variables correspond to edges in a graph, therefore the aforementioned procedure is called edge discovery.

The complexity of edge discovery upon addition of a single module is of order $O(n)$ in the number of BN modules and $O(k^2)$ in the number of variables in an individual module. The complexity of the edge discovery procedure for the assembly of n modules runs in $O(n^2k^2)$. I.e. we need, in worst case, n^2k^2 calls to the CMI measure.

A. Superfluous Edges

The set \mathcal{E} might contain also pairs which are not directly dependent according to $P(V \cup V_i)$. Such superfluous edges are not needed to restore I-mapness and we should eliminate as many as possible of such edges because:

- The model becomes unnecessarily complex and therefore difficult to modularize;
- Many superfluous parameters need to be estimated;
- Reasoning becomes computationally more demanding as the number of edges increases.

For example, consider the situation in Figure 4(a) where we want to extend $\Omega = (\{\psi_1\}, \emptyset)$ with ψ_2 to get $\Omega^* = (\{\psi_1, \psi_2\}, \{\langle \psi_2, \psi_1 \rangle\})$. Applying the edge discovery described in Proposition 2, the new modular BN Ω^* will result in the following undirected edges $\mathcal{E} = \{(M, C_1), (M, R_1), (Smell, C_1), (Smell, R_1)\}$, because the four pairwise CIs which assumed $(M \perp\!\!\!\perp C_1 | Cond)_P$, $(M \perp\!\!\!\perp R_1 | Cond)_P$, $(Smell \perp\!\!\!\perp C_1 | Cond)_P$ and $(Smell \perp\!\!\!\perp R_1 | Cond)_P$ failed the test (see ground truth model in Figure 1 why these pairwise CIs fail). The dependence between the variables M and C_1 (dashed directed edge between M and C_1 in Figure 3) induces dependencies between

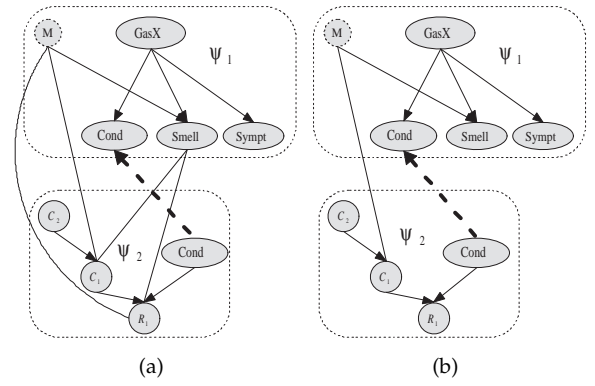


Figure 5. Discovering the structure of the gas detection fusion model

variable pairs (M, R_1) , $(R_1, Smell)$ and $(C_1, Smell)$ corresponding to superfluous edges shown in Figure 5(a).

However, in general, many of the undirected edges found by applying the *pairwise CI* test are often conditionally independent given a conditioning set that differs from the conditioning set used in the pairwise CIs. Therefore, for every variable pair in an edge $E = (X, Y)$, where $E \in \mathcal{E}$, we try to find a set S of variables that make the two variables X and Y conditionally independent. Failing to find such a set justifies an edge between X and Y . However, if we find a set S which makes the two variables independent given S then the edge (X, Y) is superfluous and must be removed from the graph.

To find the conditioning set S efficiently for a variable pair (X, Y) we use a modified version of the *PC-algorithm* [4] which works directly on the undirected edges found by exhaustive testing of *pairwise CIs*. The complexity of the PC-algorithm in worst case is exponential in the maximum degree of any variable in the domain. However, in practice this is rare [4]. The modified version of the PC-algorithm is more economical than the original PC-algorithm, because we only have to consider the undirected edges that were found by *pairwise CIs* tests upon addition of a module. Consequently, there is no need to run the PC-algorithm on a complete undirected graph and therefore less variables have to be considered and the degree of the variables will be smaller.

After eliminating the superfluous edges the undirected edges between directly dependent variables remain. These undirected edges are oriented (add directionality) by exploiting converging connections [4], [5]. Such connections have a unique statistical signature and can therefore be used to discriminate between the different types of connections (see Definition 1) and consequently add orientation to *some* of the remaining undirected edges. Next to the converging connections a DAG also has serial and diverging connections. Because the serial and diverging connections have an identical statistical signature we can only restore the representation of the dependencies up to the *Markov equivalence class* [9]. However, this class includes the ground truth model.

| M | C_2 | | C_1 | |
|--------------|--------------|------------|-------------|--------------|
| | <i>high</i> | <i>low</i> | <i>high</i> | <i>low</i> |
| $C_1 = ok$ | α | 0.99 | 0.2 | $1 - \alpha$ |
| $C_1 = fail$ | $1 - \alpha$ | 0.01 | 0.8 | α |

Table I
CPT USED FOR THE POSTERIOR ANALYSIS.

The model in Figure 5(b) is not correctly modularized after learning the dependence between C_1 and M . For this specific example repairing the structure can be accomplished by using the design rules for modular Bayesian networks proposed in [2]. This requires duplication of variable M in the module ψ_2 as shown in Figure 2.

B. Dealing with Limited Data Sets

Empirically, $I(\mathbf{X}, \mathbf{Y} | \mathbf{Z})$ will never be exactly 0, even if a finite data set \mathcal{D} is sampled from a distribution for which $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_p$. Therefore, a threshold δ is needed for independence tests. In our approach δ is based on the conditional mutual information corresponding to pairs of variables in local models for which we know that are independent. We know that because of the local I-map assumption (see Definition 3): the inner BN module d-separations correspond to valid conditional independences. Consequently, the threshold δ is defined as: $\delta = K \max_j I_j(\mathbf{X}, \mathbf{Y} | \mathbf{Z}) \in \mathbf{I}^M$, where \mathbf{I}^M is a set of computed CMI measures corresponding to d-separations in BN modules of M and K is a multiplication factor.

VI. EXPERIMENTS

In this section we experimentally show (i) the impact of inadequate representation of dependencies and (ii) detection of faulty modular BNs and discovery of crucial dependencies which are required to restore I-mapness.

A. Posterior Analysis

We assume that the humidity variable M has a causal influence on the sensor components C_1 , C_3 and $Smell$ (see Figure 1). In the modular BN in Figure 3 the dependence between the humidity variable M and the sensor component C_1 is not captured. Consequently, the modular BN does not correspond to the correct I-map shown in Figure 1. We investigate the effect of this violation on the posterior $P(GasX | R_1 = true, R_{Smell} = true, M = low)$ computed in the modular BN given in Figure 3 and compare it with the posterior obtained with the help of the correct modular BN in Figure 2. The missing relation in the faulty modular BN is reflected in the CPT $P(C_1 | C_2)$, which is computed from the CPTs of the ground truth distribution represented by the network in Figure 1. Variable α in the CPT for $P(C_1 | C_2, M)$ (see Table I) is used to variate the strength of influence of humidity M on the sensor component C_1 ³.

The results are plotted in Figure 6. The solid curve shows posterior $P(GasX | R_1 = true, R_{Smell} = true, M = low)$ computed from the correct modular BN in Figure 2. The dashed curve shows the posterior $P(GasX | R_1 = true, R_{Smell} = true, M = low)$ computed from the incorrect modular BN in Figure 3. These curves show that inadequate representation of dependencies can have a

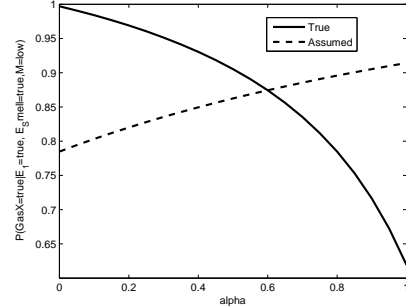


Figure 6. The effect on the posterior $P(GasX = true | R_1 = true, R_{Smell} = true, M = low)$ as a function of α (see Table I)

significant effect on the posterior. The difference between the posterior computed from the true and the incorrect modular BN depends on the given true distribution (see α in Table I).

B. Dependence Discovery

In the following experiment we analyze the performance of the presented model verification method described in Sections IV and V.

We assume again a partially faulty distributed model from Figure 3, which does not correctly capture relations between variables M and C_1 . The data is generated through sampling from the true network in Figure 1. For this experiment the data size is varied and the performance on discovering the dependence between M and C_1 is measured. For each data size the experiment is repeated 50 times. For the CPT in Table I we used $\alpha = 0.75$.

In Figure 7 the results are plotted. The solid curve represents the percentage of exactly discovering the dependence (including edge orientation) between M and C_1 . The dashed curve shows the percentage of cases in which the edge between M and C_1 was discovered but also superfluous edges were returned. The dotted curve shows the percentage of cases in which we could not discover the dependence between M and C_1 at all. From these curves it is clear that the performance increases when we increase the size of the data. By using 300 samples or more, we could discover the dependence between M and C_1 in more than 90% of the cases. In addition, the remaining cases were mostly situations where we discovered the dependence between M and C_1 , but also superfluous edges. In such situation we are still able to repair the fusion model and do correct

³The specification of the remaining CPTs for this experiment are omitted because of the lack of space.

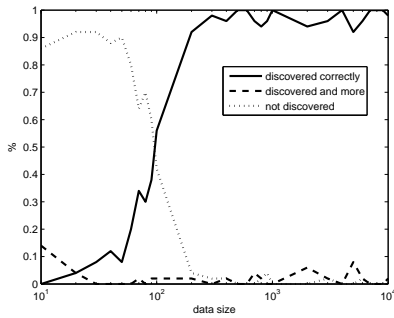


Figure 7. Results discovery of the dependence between M and C_1 in the modularized gas detection fusion model given in Figure 3

reasoning. Only a very small percentage of the cases the dependence between M and C_1 could not be discovered.

VII. DISCUSSION

This paper introduces an approach to runtime inspection and identification of dependencies in modular Bayesian fusion systems. The approach supports verification of the modeled dependencies between variables by combining (i) information theoretic analysis of data samples and (ii) prior domain knowledge captured by fusion modules. It is particularly suitable for verification of Bayesian causal models in distributed fusion systems, which are gradually extended with new fusion modules. In the context of monitoring systems we assume that correct local models of sensor devices are known. This assumption is realistic since conditional probabilities over a limited set of variables representing a sensor device can be obtained via controlled experiments.

Nevertheless, the data-driven analysis based on pairwise CI tests requires acquisition of sufficient amounts of data. Beside easily accessible reports from sensors and humans, the analysis also requires observations of variables which are hidden during normal operation. This is the case with the variables we condition on in the pairwise CI tests. Moreover, in general settings validation requires CI tests on pairs involving also variables which are not observed during normal operation. States of such hidden variables could be observed by using well calibrated, high quality sensors in the evaluation phase, prior to putting a new type of modules into regular service. For each sensor type we could equip a single sensor device with additional sensors observing its internal states. For example, special sensors could be used for the observation of the internal states of sensor components corresponding to model variables C_1 , C_2 in Figure 1. If we used a single *fully observable* sensor device for each type of sensors we could obtain all the required data for the verification of the corresponding fusion modules and identify missing links between modules. Note that we test the combination of types of sensors and not a particular constellation of sensor devices. Similarly, in

case of reports from citizens, chemical advisers proficient in recognizing smell could provide reliable observations about the variable *Smell* and medical staff could provide reliable information about the symptoms.

Clearly, observation of states which are normally hidden at runtime can be expensive. Thus, the question is whether the acquisition of sufficient quantities of observations is economically justified. The number of required observations can be kept relatively low if a new module is added to the fusion system if and only if the evaluation tests indicate no violation of dependencies in the context of distributed system. In this way we achieve that the distributed fusion systems correctly capture relations between variables dispersed throughout all fusion modules. In such cases we know that a new module correctly captures relations w.r.t. an existing distributed fusion system if (i) we form all possible pairs between the variables from the new module and the variables in the existing modules and (ii) all pairs pass the CI test. Such validation effort is justified in applications, where many fusion modules of each type will be used; the full observability is required in the initial phase which involves a single module of each type. Moreover, *discovery* of missing dependencies in a distributed model of a monitoring process is in principle more expensive than detection/verification of modules. Namely, it requires observations of all variables as well as variation of the sets of variables which are used for the conditioning. However, even such identification of ignored relations (links) between partial causal models is more efficient than common approaches to structure learning from scratch.

While we used DPN-based fusion systems for illustration, the presented method could be used with any approach to distributed Bayesian inference, such as for example [7].

REFERENCES

- [1] G. Pavlin, P. de Oude, M. Maris, J. Nunnink, and T. Hood, "A distributed approach to information fusion based on causal probabilistic models," Tech. Rep., 2007.
- [2] P. de Oude, G. Pavlin, and T. Hood, "A modular approach to adaptive bayesian information fusion," in *In The 10th International Conference on Information Fusion*, Quebec City, Quebec, Canada, 2007.
- [3] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [4] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*. The MIT Press, 2000.
- [5] J. Pearl, *Causality: Models Reasoning and Inference*. Cambridge University Press, 2000.
- [6] F. V. Jensen, *Bayesian Networks and Decision Graphs*. New York: Springer-Verlag, 2001.
- [7] Y. Xiang, *Probabilistic Reasoning in Multiagent Systems: A Graphical Models Approach*. Cambridge University Press, 2002.
- [8] A. P. Dawid, "Conditional independence in statistical theory," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 41, pp. 1–31, 1979.
- [9] R. Neapolitan, *Learning Bayesian Networks*. Prentice Hall, 2003.