A Modular Approach to Adaptive Bayesian Information Fusion

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

Gregor Pavlin Thomas Hood
Thales Research & Technology Netherlands
Delftechpark 24, 2628 XH, Delft, The Netherlands
Email: {gregor.pavlin,thomas.hood}@icis.decis.nl

Abstract*—In this paper we show that causal probabilistic models can facilitate the design of robust and flexible fusion systems. Observed events resulting from stochastic causal processes can be modeled with the help of causal Bayesian networks, mathematically rigorous and compact probabilistic causal models. Bayesian networks explicitly represent conditional independence which facilitates decentralized modeling and information fusion. Starting with the theory of BNs we derive design and organization rules for distributed multi-agent systems that implement exact belief propagation without centralized configuration and fusion control. In this way we can design multi-agent fusion systems which can adapt to rapidly changing information source constellations and can efficiently process large quantities of information.

Keywords: distributed fusion, heterogeneous information, multi agent systems, Bayesian networks.

I. Introduction

Contemporary decision making processes require accurate and quick situation assessment. Particularly challenging and increasingly relevant is situation assessment in crisis management processes, 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).

Interpretation of different types of information requires adequate domain models which provide a mapping between heterogeneous observations and hypotheses about hidden events. However, situation assessment in contemporary decision making and control settings introduces several substantial challenges:

- Domains are often complex, which means that models are abstractions associated with significant uncertainties.
- Information sources are heterogeneous and noisy. The heterogeneity implies complex domain models.
- Constellations of information sources are often not known prior to the operation and they change at runtime.

It turns out that causal Bayesian networks (BN) [1], facilitate design of robust and flexible fusion systems which can cope with these challenges. Namely, monitoring processes can often be viewed as causal stochastic processes, where hidden events cause observations with certain probability. In such settings, BNs provide a theoretically rigorous and compact causal mapping between hidden events of interest and observable events. However, such models must capture every information source and observation explicitly. Thus, each fusion process requires a specific domain model which maps observations from the current constellation of information sources to the hypotheses of interest. In addition, large numbers of information sources require large BNs, which in turn might require significant communication and processing resources. We can cope with these problems by introducing modular fusion systems. In such systems, adequate causal domain models can be assembled out of basic building blocks as information sources become available. In addition, a modular approach supports distribution of partial fusion processes throughout a network of processing devices. In this way we can avoid communication and processing bottlenecks.

Causal BNs facilitate decentralized world modeling and information fusion, because they explicitly capture conditional independence which is reflected in the factorization of the joint distributions represented by such BNs [2]. Moreover, in monitoring processes each event is significantly influenced only by a small fraction of other events. Such locality of causal relations and the corresponding factorization provide a guidance for efficient distribution of models and fusion processes. In particular, we emphasize the use of Markov boundaries [3], [4] which can render subsets of variables within a BN conditionally independent of each other[1].

By using Markov boundaries, we derive theoretically rigorous rules for systematic transformation of monolithic BNs into equivalent distributed models which support efficient and correct belief propagation; i.e. given a certain set of observations, the monolithic or distributed approaches yield identical results. The resulting distributed fusion systems support theoretically correct fusion without any compilation of secondary probabilistic structures (e.g. Junction trees [5]) spanning several modules (i.e. agents) and no centralized control of the fusion and configuration processes is needed. In other words, we can cope with ad-hoc information source constellations efficiently.

The presented modeling principles are used in *Distributed*

[1]If all variables in a Markov boundary are instantiated (i.e. their states are observed), then the inference based on variables within this boundary is independent of inference in the rest of the model.

Perception Networks (DPN), which are especially suited for distributed information fusion in monitoring applications with ad-hoc sensory infrastructure. The DPNs are multi agent systems (MAS) [6], where agents implement cooperating processing nodes with limited fusion capabilities based on local BNs (i.e. limited domain knowledge). Beside theoretically sound information fusion, DPNs implement communication and cooperation protocols which support self-configuration of arbitrarily large distributed fusion systems. The resulting fusion systems compute correct (exact) posterior probabilities over certain variables of interest by using evidence obtained from different cooperating agents. DPNs support flexible distributed fusion systems since the local BNs are designed in such a way that dependencies between different local fusion processes from different agents are minimized.

Our approach is complementary to other approaches to belief propagation in distributed BNs [5], [7]. Contrary to our approach, these approaches require global secondary structures for inference or partial coordination of global processes (see Section IV).

This paper provides a theoretical rationale for new design principles for Bayesian fusion in MAS. The presented principles were applied to the design of different demonstrators (see for example [8]).

II. State Estimation with Causal Bayesian Networks

We assume that situations can be described through finite sets of discrete random variables. For example values of binary variables could represent the presence/absence of fire, smoke, etc. Moreover, we assume that hidden and observed states of the domain, such as the presence/absence of smoke or a report from a smoke detector, materialize through 'sampling' from some true distribution over the combinations of possible states. Thus, in a particular situation certain possible states materialize while others do not. In other words, we assume that the ground truth corresponds to a point mass distribution over the possible states of discrete random variables in the targeted domain. Consequently, state estimation is classification of the possible combinations of relevant states. Given a discrete hypothesis variable H and a set of observations $\mathcal{E}$, the estimation corresponds to the determination of the hypothesis $\hat{h}$ for which the posterior probability $P(H|\mathcal{E})$ or, equivalently, the joint probability $P(H,\mathcal{E})$, is maximum: $\hat{h} = argmax_{h_i}(P(h_i,\mathcal{E}))$. Thus fusion corresponds to the computation of the joint distribution $P(h_i,\mathcal{E})$ over the states h_i and $\mathcal{E}$ for every possible h_i.

If the observable states are outcomes of stochastic causal processes then $P(h_i,\mathcal{E})$ or $P(h_i|\mathcal{E})$ can be computed with the help of causal Bayesian networks (BN) in a theoretically rigorous and compact way [1]. A Bayesian network is defined as a tuple $\langle \mathcal{G}, P\rangle$, where $\mathcal{G} = \langle \mathcal{V}, \mathbb{E}\rangle$ is a directed a-cyclic graph (DAG) defining a domain $\mathcal{V} = \{V_1,\ldots,V_n\}$ and a set of directed edges $\langle V_i, V_j\rangle \in \mathbb{E}$ over the domain. The joint probability distribution $P(\mathcal{V})$ over the domain $\mathcal{V}$ is defined as $P(\mathcal{V}) = \prod_{V_i\in\mathcal{V}} P(V_i|\pi(V_i))$, where $P(V_i|\pi(V_i))$ corresponds

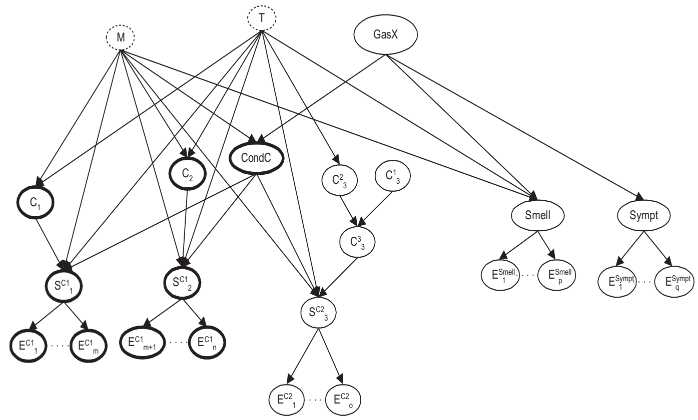

Figure 1. A causal model capturing relations between the possible states, such as presence/absence of GasX represented by node $GasX$ and different types of sensor observations. Highlighted portion of the graph captures causal relations between $CondC$, sensor propensities S_i^{C1} and sensor reports E_j^{C1}.

to the conditional probability table (CPT) for node V_i given its parents $\pi(V_i)$ in the graph. In this paper we assume that each node represents a discrete variable. In general, the probability distribution over an arbitrary set of discrete variables can be computed through marginalization of $P(\mathcal{V})$. BNs in general can be used as models describing probabilistic relations among different events. Given a BN we can choose a query node or hypothesis node H with values h_i and compute probability distribution $P(H|\mathcal{E})$ over H for a given evidence pattern $\mathcal{E}$. Evidence $\mathcal{E}$ corresponds to a certain constellation of variable instantiations (also configuration) and subsequent inference results in a distribution $P(H|\mathcal{E})$ which determines a "score" $P(h_i|\mathcal{E})$ for each hypothesis $h_i \in H$.

A. Causal Domain Models of Monitoring Systems

Bayesian networks are relevant for a significant class of monitoring domains, where we can identify causal relationships between hidden and observed events[2]. We illustrate this with the help of a simple gas detection system that makes use of two different types of sensors as well as reports from humans (see the causal model in Figure 1). The existence of $GasX$[3] 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 in conjunction with thermometers and hygrometers. 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 $CondC$ correspond to the situations where, under ideal circumstances,

[2]In this paper an event is synonymous to a realization of certain states

[3]For the sake of simplicity we say that a toxic gas is present if its concentration exceeds some critical value. Otherwise we say that the gas is absent.

electrical current in an exposed semiconductor would either exceed some detection threshold (i.e. $CondC = true$) or be below that threshold (i.e. $CondC = false$). The current will exceed the detection threshold with the probability $P(CondC = true|GasX = true)$ if the gas concentration exceeded a critical value, i.e. $GasX = true$. Also, with probability $P(CondC = false|GasX = true)$ the current might not exceed the detection threshold despite $GasX = true$. This might be due to the presence of another gaseous component which would inhibit the desired chemical reaction between $GasX$ and the semiconductor.

Moreover, we assume that $GasX = true$ within a finite time interval, i.e. time slice. Within such a time slice electronic circuitry of a sensor evaluates the current and generates a stream of sensor reports. Let the jth report from a sensor of type i be represented by a random variable E_j^i. A sensor report claiming the presence or absence of some hypothetical phenomenon is expressed by $E_j^i = true$ or $E_j^i = false$, respectively. Given the existence of a critical gas concentration, the ith sensor is considered to be working correctly if $P(E_j^i = true|GasX = true)$, the probability that a report from this sensor asserts the presence of the gas, is greater than $P(E_j^i = false|GasX = true)$, the probability that a report asserts the absence of the gas. Similarly, given the absence of a critical gas concentration, the relation $P(E_j^i = true|GasX = false) < P(E_j^i = false|GasX = false)$ should be satisfied when a sensor is considered to be working correctly. In other words, if a sensor works correctly the majority of the sensor reports should assert the presence or absence of $GasX$ given $GasX = true$ or $GasX = false$, respectively. Nodes $C_1, \dots, C_3^3, C_3^2, C_3^1$ represent states of critical electronic components in the three sensors.

However, probability distribution $P(E_j^i|GasX)$ does not depend only on the presence of the phenomenon that we are trying to infer and the states of the electronic components. For example, the sensor electronics might fail due to proximity to a source of electromagnetic noise, the sensor could be covered by ice, etc. In other words, the true distribution over reports depends on many factors which are often not well known. We avoid detailed modeling of causal processes producing sensor reports by introducing the *sensor propensity* concept that represents two types of situations, each corresponding to certain combinations of the states of the electronic components and the states in the domain (e.g. conductivity of the chemical element) [9]. We represent the *sensor propensity* by a binary variable S_i. $S_i = true$ denotes the class of state combinations which influence the sensing process, such that the probability of obtaining a sensor report affirming the existence of some phenomenon is greater than 0.5; i.e. $P(E_j^i = true|S_i = true) > P(E_j^i = false|S_i = true)$. Situations corresponding to $S_i = false$ would influence the distribution over types of sensor reports, such that $P(E_j^i = true|S_i = false) < P(E_j^i = false|S_i = false)$.

Besides the semiconductor sensors we assume that there are humans in the area who have olfactory reactions to $GasX$ and who submit reports of what they smell via a call service. These reports are represented by the variable $Smell$. Smelling capabilities can be influenced by the temperature and humidity denoted by variables T and M, respectively. Such influences are captured by the conditional probability distribution $P(Smell|GasX, M, T)$. Similarly, aid workers might be able to report about health symptoms which are typical results of exposure to $GasX$ which is denoted by variable $Sympt$.

A BN describing causal relations between the aforementioned variables is depicted in Figure 1 which corresponds to the following factorization of the joint probability distribution $P(\mathcal{V})$:

$$\begin{aligned}
P(\mathcal{V}) = {} & P(GasX)P(M)P(T)P(CondC|GasX, M) \\
& \cdot P(Smell|GasX, T, M)P(Sympt|GasX) \\
& \cdot P(S_1^{C1}|CondC, T, M, C_1)P(C_1|M, T) \\
& \cdot P(S_2^{C1}|CondC, T, M, C_2)P(C_2|M, T) \\
& \cdot P(S_3^{C2}|CondC, T, M, C_3^3)P(C_3^2|T)P(C_3^3|C_3^1, C_3^2) \\
& \cdot P(C_3^1) \prod_{E_j^{C1} \in \mathcal{E}_1} P(E_j^{C1}|S_1^{C1}) \prod_{E_k^{C1} \in \mathcal{E}_2} P(E_k^{C1}|S_2^{C1}) \\
& \cdot \prod_{E_l^{C2} \in \mathcal{E}_3} P(E_l^{C2}|S_3^{C2}) \prod_{E_m^{Smell} \in \mathcal{E}_4} P(E_m^{Smell}|Smell) \\
& \cdot \prod_{E_n^{Sympt} \in \mathcal{E}_5} P(E_n^{Sympt}|Sympt), \qquad (1)
\end{aligned}$$

where $\mathcal{E}_1, \dots, \mathcal{E}_5$ denote sets of evidence, from sensors and humans, corresponding to instantiations in different parts of the BN. The total evidence set is denoted by $\mathcal{E} = \bigcup_{i=1}^{5} \mathcal{E}_i$. Also, the third and the fourth line correspond to two sensors of the same type. Consequently, $P(S_1^{C1}|CondC, T, M, C_1) = P(S_2^{C1}|CondC, T, M, C_2)$ and $P(C_1|M, T) = P(C_2|M, T)$. Similarly, $P(E_j^{C1}|S_i^{C1})$ are identical for all j.

Note that with each sensor we introduce an independent partial causal process which is initiated through some hidden phenomenon. For example, by introducing a new gas sensor the presence of gas will initiate different processes in the sensor's circuitry which will eventually produce sensor reports. This is reflected in a BN by introducing a new propensity node as well as nodes representing reports and electronic components.

B. Temporal Aspects and Context Variables

We assume that the domain's hidden phenomena are *quasi-static* in the sense that they do not change during a single time slice. For example, in Figure 1 nodes $GasX$ and $CondC$ represent the presence/absence of gas and increased conductivity of the semiconductor, respectively. If $GasX$ is present then we assume that it is present throughout the time slice; i.e. $GasX = true$ for the duration of the time slice. On the other hand, within a time slice we obtain several sensor reports at different points in time. These reports may vary throughout the time slice. For a significant class of sensors it can be assumed that the sequence of reports results from a first order Markov process. However, it has been shown that such a process can be modeled as a set of branches rooted in a single node, provided that the hidden phenomenon is *quasi-static* (see [10]). Consequently, for each observation from a sequence, which is obtained within a single time slice, we can append a new leaf node. In addition, models can feature the so called *context variables* which influence the causal processes of interest but they can be considered independent of these processes. Therefore, context variables are always represented through root nodes and remain constant during a time slice, i.e. context variables are quasi-static. For example, Figure 1 suggests that different sensors are influenced by the air temperature and humidity represented by variables T and M, respectively. However, we can safely assume that the sensing processes do not significantly influence the temperature and humidity. The main difference between observations and context variables is that the former are outcomes of stochastic causal processes while the later are not.

C. Fusion Based on Inference in Bayesian Networks

Bayesian networks support efficient fusion of very heterogeneous information based on rigorous and efficient algorithms for the computation of probability distributions $P(H|\mathcal{E})$ or $P(H,\mathcal{E})$ for given observations $\mathcal{E}$. For example, we might be interested in distribution $P(CondC|\mathcal{E} = \{E_1^{C_1}, \ldots, E_q^5\})$ which can be obtained through inference in the BN shown in Figure 1.

In principle, the computation of $P(H|\mathcal{E})$ or $P(H,\mathcal{E})$ requires marginalization of all variables in the BN except H and the evidence variables that were instantiated according to $\mathcal{E}$. If we can express a joint probability distribution over $\mathcal{V}$ as a product of factors corresponding to conditional and prior probabilities, then the computation can be made efficient by using appropriate sequences of factor multiplications and summations over the resulting products. Marginalization can be very efficient if BNs feature tree topologies by using lambda-pi message passing algorithm [1]. Often, however, domain models are described through multiply connected BNs. This requires that the original BNs be mapped to alternative representations which facilitates correct marginalization. There basically exist two ways of doing this:

- Multiply connected BNs can be transformed into secondary representations, such as Junction trees. Basically, Junction trees collect nodes from an original BN into clusters which represent hyper nodes in a tree.
- Through instantiation of certain variables in a multiply connected BN, we can reduce dependencies, such that algorithms for simple tree topologies can be applied.

Note that computation of Junction trees and other similar secondary fusion structures in distributed systems [5], [7] requires expensive processing and massive messaging. This in turn can be impractical in domains where constellations of information sources change rapidly. In such domains we might never end up with a viable global structure spanning several agents. Therefore, in this paper we focus on distributed inference methods which do not require computation of secondary fusion structures.

D. Locality of Causal Relations

Locality of causal relations in a BN can facilitate decomposition of inference processes. An important concept is d-separation [1], [4], which describes how different parts of a BN can be rendered conditionally independent. Lets assume a *faithful*[4] DAG $\mathcal{G} = \langle \mathcal{V}, \mathbb{E} \rangle$ and mutually disjoint sets of variables $\mathcal{V}_i \subset \mathcal{V}$, $\mathcal{V}_j \subset \mathcal{V}$ and $\mathcal{Z} \subset \mathcal{V}$. Sets $\mathcal{V}_i$ and $\mathcal{V}_j$ are d-separated by $\mathcal{Z}$, if instantiation of all variables in set $\mathcal{Z}$ makes arbitrary variable $X \in \mathcal{V}_i$ conditionally independent of any variable $Y \in \mathcal{V}_j$. D-separation of $\mathcal{V}_i$ and $\mathcal{V}_j$ given $\mathcal{Z}$ in DAG $\mathcal{G}$ is denoted by $\langle \mathcal{V}_i|\mathcal{Z}|\mathcal{V}_j \rangle_{\mathcal{G}}$. In other words, if all variables in set $\mathcal{Z}$ are instantiated and $\langle \mathcal{V}_i|\mathcal{Z}|\mathcal{V}_j \rangle_{\mathcal{G}}$, then computation of probabilities over variables $\mathcal{V}_i \subseteq \mathcal{V}$ is independent of the instantiation and belief propagation over variables $\mathcal{V}_j \subseteq \mathcal{V}$.

Moreover, d-separation is related to the central concept of our analysis, the Markov boundary:

Definition 1 (Markov boundary of a single variable): Given a probabilistic model $P(\mathcal{V})$ over a set of variables $\mathcal{V}$ and a variable $X \in \mathcal{V}$, then a Markov boundary $B(X)$ of X is a minimal set of variables $\mathcal{Z}$ for which the following conditional independence is valid

$$I(\{X\}, \mathcal{Z}, \mathcal{V}\backslash(\{X\} \cup \mathcal{Z}))_P. \tag{2}$$

where $I(\{X\}, \mathcal{Z}, \mathcal{V}\backslash(\{X\} \cup \mathcal{Z}))_P$ denotes that X is conditionally independent of any variable in the set $\mathcal{V}\backslash(\{X\} \cup \mathcal{Z})$ given the set $\mathcal{Z}$.

Moreover, it can be shown that Markov boundary $B(X)$ of any variable X in a *faithful* DAG is the union of all parents of X, children of X and parents of children of X (see [4] for the proof).

We can extend the concept of Markov boundary to a set of variables. This is based on the concepts of parents and children of a set of variables [3]:

Definition 2: Let $\mathcal{G}$ be a DAG over a set $\mathcal{V}$ of variables. Let $\mathcal{V}_i$ be a set of variables such that $\mathcal{V}_i \subset \mathcal{V}$ and $\mathcal{V}_i \neq \emptyset$. Any node $X \in \mathcal{V}\backslash\mathcal{V}_i$ that has a child in $\mathcal{V}_i$ is a parent of set $\mathcal{V}_i$; any node in $X \in \mathcal{V}\backslash\mathcal{V}_i$ that has a parent node in $\mathcal{V}_i$ is a child of set $\mathcal{V}_i$.

By using parents and children of sets of variables we can define a Markov boundary of a set of variables [3]:

Definition 3: Markov boundary $B(\mathcal{V}_i)$ of a set of variables $\mathcal{V}_i \subset \mathcal{V}$ in a faithful *DAG is the union of all parents of set $\mathcal{V}_i$, children of $\mathcal{V}_i$ and parents of children of $\mathcal{V}_i$.*

Markov boundary of a set of variables $B(\mathcal{V}_i)$ implies that computation of beliefs of any of the variables in $\mathcal{V}_i$ can be carried out in isolation, independently of inference processes in the rest of the network for variables $\mathcal{V}\backslash\{\mathcal{V}_i \cup B(\mathcal{V}_i)\}$, if all variables in $B(\mathcal{V}_i)$ are instantiated. This is a very important property which can be exploited for efficient distribution of models and inference processes (see section III-B).

[4]Given a probabilistic model $P(\mathcal{V})$ and a DAG $\mathcal{G}$, we say that P and $\mathcal{G}$ are faithful if $\mathcal{G}$ entails all conditional independencies in P as well as all dependencies, i.e. $\mathcal{G}$ is a perfect map of P (see [4])

III. DISTRIBUTED FUSION

While BNs can cope with modeling and observation uncertainties, in contemporary fusion problems we are often confronted with additional challenges. As we already pointed out in the introduction, constellations of information sources change at runtime. However, every information source must be captured in a BN. Thus for each constellation of information sources we need a specific BN. In addition, large quantities of heterogeneous information accessed through the existing communication and sensing infrastructure often require large BNs which in turn require significant processing and communication resources.

These problems can be tackled by introducing modular approaches to modeling and processing. By using basic fusion building blocks, each specialized for a limited fusion task, adequate domain models (i.e. BNs) can be assembled at runtime as the information sources are discovered. In addition, fusion in such an assembled network can easily be distributed throughout several machines thus avoiding processing and communication bottlenecks. Beside sound and efficient fusion algorithms, basic fusion modules must support also efficient communication and cooperation protocols. In addition, a distributed fusion system should be able to adapt to the current situation without intervention of humans, which requires autonomous behavior; modules should form fusion systems consisting of relevant modules autonomously and they should be able to reason about resource allocation with respect to sensing and processing capacity. In order to be able to cope with such complex functionality in a systematic way, we make use of the multi agent systems paradigm [6].

Definition 4 (Fusion Agent): **Fusion agent** *A_i is a processing unit, a module, which can compute probability distributions over variables in its local BN.*

- *A local BN is defined through a local DAG $\mathcal{G}_i = \langle \mathcal{V}_i, \mathbb{E}_i \rangle$ and a set of conditional probabilities which encode factorization of a joint probability distribution $P(\mathcal{V}_i)$ over local variables $\mathcal{V}_i$.*
- *Each agent A_i maintains a set of **service variables** $\mathcal{R}_i \subset \mathcal{V}_i$ and a set of **input variables** $\mathcal{L}_i \subset \mathcal{V}_i$.*
- *Each agent A_i can compute marginal posterior probabilities over the service variables and communicate this result to other agents that are interested in this distribution, i.e. agents that have the service variable as input variable.*

Two agents can establish a fusion contract and exchange local estimates of marginal distributions for any variable contained in their separator:

Definition 5 (Separator): Given two variable clusters $\mathcal{V}_i$ and $\mathcal{V}_j$ corresponding to agents A_i and A_j, the separator $S\langle \mathcal{V}_i, \mathcal{V}_j \rangle$ is defined by $S\langle \mathcal{V}_i, \mathcal{V}_j \rangle = \mathcal{V}_i \cap \mathcal{V}_j$

A. Fusion Organization

Each fusion process depends on the constellation of cooperating agents, which corresponds to a particular problem/task decomposition. In this context we use the concept of a *Fusion Organization* [9]:

Definition 6 (Fusion Organization): Fusion Organization is defined by a function $\mathcal{F}(H, \mathcal{A}) : \mathcal{A}, H \rightarrow \Omega_t$, which maps the set of existing agents $\mathcal{A}$ and a query concept (hypothesis variable) H, i.e. a hidden variable of interest, to a graph Ω_t at timestep t. Ω_t is a tuple $\Omega_t = \langle \mathcal{A}_H, \mathcal{C}_H \rangle$, where $\mathcal{A}_H \subseteq \mathcal{A}$ denotes the set of agents that can provide information relevant for the reasoning about the distribution over the variable corresponding to the query concept H. $\mathcal{C}_H$ represents a set of agent pairs with common separators (see Definition 5).

A global task of a particular fusion organization Ω_t is computation of probability distribution $P(H|\mathcal{E})$ over some hypothesis variable $H \in \mathcal{R}_i$ of agent A_i which correctly reflects the entire evidence set $\mathcal{E}$. This is the case if $P(H|\mathcal{E}) = P'(H|\mathcal{E})$, where $P'(H|\mathcal{E})$ is computed through propagation of the entire evidence $\mathcal{E}$ in a monolithic BN which correctly captures all dependencies between the variables contained in the local models of agents from Ω_t. Correct distributed computation requires well defined cooperation of fusion agents since evidence $\mathcal{E}$ corresponds to instantiations of variables in different agents in a fusion organization Ω_t.

In the following section we investigate how local BNs of different fusion agents should be determined in order to obtain agent organizations which support correct computation of distributions over hidden variables without (i) any global compilation of secondary fusion structures (e.g. Junction trees) and (ii) without centralized fusion control[5]. We show that this is possible, if local BNs in different agents satisfy simple conditions and variables are instantiated in a certain way.

B. Model Decomposition

Given an arbitrarily complex monolithic BN with a DAG $\mathcal{G} = \langle \mathcal{V}, \mathbb{E} \rangle$, we introduce a system of fusion agents such that each agent A_i is reasoning about a subset of variables $\mathcal{V}_i \subset \mathcal{V}$ from $\mathcal{G}$. Local BNs with DAGs $\mathcal{G}_i = \langle \mathcal{V}_i, \mathbb{E}_i \rangle$ are obtained through partitioning of a monolithic BN. Each local BN corresponds to a local domain expertise of an agent A_i.

Moreover, in a multiply connected monolithic BN we can eliminate dependencies between subsets $\mathcal{V}_i \subset \mathcal{V}$ through instantiation of variables. In order to be able to systematically determine which variables should be instantiated, we consider properties of Markov boundaries. Recall, that if all variables in the Markov boundary $B(\mathcal{V}_i)$ are instantiated, then the computation of beliefs in one subset $\mathcal{V}_i \subset \mathcal{V}$ can be carried out in isolation, independently of inference processes in the rest of the network corresponding to variables $\mathcal{V} \backslash \mathcal{V}_i$.

We first introduce rules for the generation of local BNs from a monolithic BN, which guarantee that all dependencies between different local graphs are preserved.

Definition 7 (Design Rules): Let DAG $\mathcal{G}$ of a monolithic BN be defined over a set of variables $\mathcal{V}$. From this monolithic BN we obtain local DAGs $\mathcal{G}_i$ by applying the following rules:

1) *DAG $\mathcal{G}$ is partitioned into n subgraphs $\mathcal{G}_1^*, \ldots, \mathcal{G}_n^*$ defined over sets of variables $\mathcal{V}_1^*, \ldots, \mathcal{V}_n^*$, such that*

[5] Central processes which determine messaging sequences between agents.

$\mathcal{V} = \bigcup_i \mathcal{V}_i^*$ *and* $\forall i \neq j : \mathcal{V}_i^* \cap \mathcal{V}_j^* = \emptyset$[6]. *All edges between the variables in* $\mathcal{V}$ *which are also in* $\mathcal{V}_i^*$ *are also contained in* $\mathcal{G}_i^*$ *for all* i.

2) *A local DAG* $\mathcal{G}_i$ *is obtained by augmenting* $\mathcal{G}_i^*$ *with all parents* $\pi(\mathcal{V}_i^*)$ *of set* $\mathcal{V}_i^*$. *Note that, in the local DAG* $\mathcal{G}_i$ *links between the nodes* $\pi(\mathcal{V}_i^*)$ *are omitted if they exist in the original monolithic BN [9].*
3) *For each root node* X *in a local DAG* $\mathcal{G}_i$, *assign a uniform prior distribution if* X *was added to the original partition* $\mathcal{G}_i^*$ *(see step 2); i.e.* X *is one of the parents of* $\mathcal{V}_i^*$: $X \in \pi(\mathcal{V}_i^*)$.

Figure 2 shows a system of local BNs which were generated from the monolithic BN from Figure 1 by using the rules from Definition 7. In this case the original BN was first partitioned into $\mathcal{V}_1^* = \{GasX, CondC, Smell, Sympt, T, M\}$, $\mathcal{V}_2^* = \{C_1, C_2, C_3^1, C_3^2, C_3^3, S_1^{C1}, S_2^{C1}, S_3^{C2}\}$, $\mathcal{V}_3^* = \{E_1^{smell}, \ldots, E_p^{smell}\}$, $\mathcal{V}_4^* = \{E_1^{sympt}, \ldots, E_q^{sympt}\}$, $\mathcal{V}_5^* = \{E_1^{C1}, \ldots, E_m^{C1}\}$, $\mathcal{V}_6^* = \{E_{m+1}^{C1}, \ldots, E_n^{C1}\}$ and $\mathcal{V}_7^* = \{E_1^{C2}, \ldots, E_o^{C2}\}$. These sets were used for the construction of local graphs shown in Figure 2.

Proposition 1: By removing from $\mathcal{G}_i$ *all parents of* $\mathcal{G}_i^*$ *and nodes that have children outside of the original partition* $\mathcal{G}_i^*$ *we obtain a new subgraph* $\mathcal{G}_i'$ *with* $\mathcal{V}_i'$. *Set* $\mathcal{V}_i \backslash \mathcal{V}_i'$ *is* $\mathcal{V}_i'$*s Markov boundary* $B(\mathcal{V}_i')$. *See the proof in [9].*

In other words, by using the design rules, we find local graphs $\mathcal{G}_i$ in which we can trivially identify a *minimal* set of variables whose instantiation renders the computation in the original partitions $\mathcal{G}_i^*$ *independent* of other variables in the original monolithic graph $\mathcal{G}$. Note that the variables in the resulting Markov boundaries $B(\mathcal{V}_i')$ are shared between different partitions $\mathcal{G}_i$, while the variables within a $B(\mathcal{V}_i')$ are used only in $\mathcal{G}_i$.

Note also an important property of local BNs obtained by using the rules from Definition 7: a nonempty separator $S\langle \mathcal{V}_i, \mathcal{V}_j \rangle = \mathcal{V}_i \cap \mathcal{V}_j \neq \emptyset$ between any two local DAGs $\mathcal{G}_i$ and $\mathcal{G}_j$ is necessarily a d-sepset (see [7] for a definition). We know from [7] that $\mathcal{V}_i \backslash S\langle \mathcal{V}_i, \mathcal{V}_j \rangle$ and $\mathcal{V}_j \backslash S\langle \mathcal{V}_i, \mathcal{V}_j \rangle$ are d-separated if and only if $S\langle \mathcal{V}_i, \mathcal{V}_j \rangle$ is a d-sepset.

By leaving uninstantiated variables in $B(\mathcal{V}_i')$ we allow a flow of information (local beliefs) between agents that have common variables. In this way local inference processes become dependent. However, we can limit dependencies and avoid inefficient inter-agent synchronization of local inference processes with the help of simple assembly and instantiation rules:

Definition 8 (Assembly Rules): Fusion Organization Ω_t *consisting of agents with local BNs obtained by using design rules from Definition 7 is formed as follows:*

1) *In any nonempty intersection of Markov boundaries* $B(\mathcal{V}_i') \cap B(\mathcal{V}_j'), j \neq i$ *all variables but one are instantiated.*
2) *A system of* n *cooperating fusion agents is assembled through a sequence of expansion steps. Agent* A_i *can join the current fusion organization* Ω_t *if there exists exactly one agent* $A_j \in \Omega_t$, *such that intersection* $B(\mathcal{V}_i') \cap B(\mathcal{V}_j') \neq \emptyset$ *and this intersection contains exactly one uninstantiated variable (see IV for a discussion); i.e. agent* A_j *already included in the current fusion organization* Ω_t *reasons about a subset of variables which are relevant also for agent* A_i *and enhanced fusion organization* Ω_{t+1} *is obtained as a new fusion contract is established between* A_i *and* A_j.

[6]Note that, we make no claims about how an optimal partitioning of a monolithic graph should be obtained.

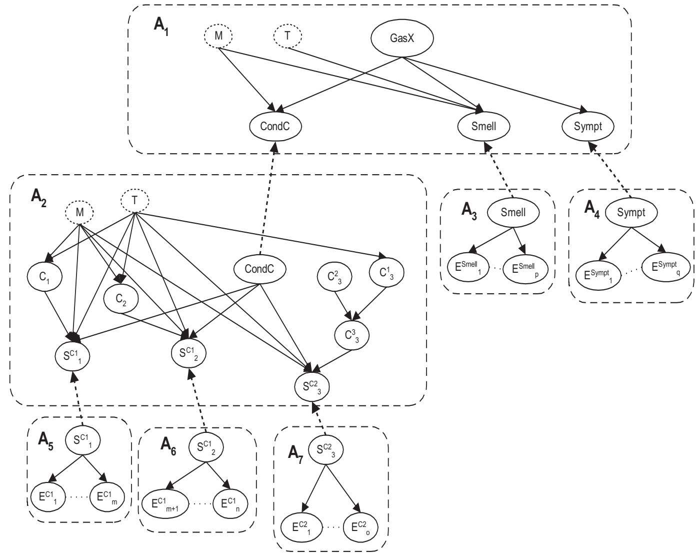

Figure 2. A distributed BN which supports inference that is equivalent to inference in the original monolithic BN shown in Figure 1. Each dashed rectangle is a local BN featuring DAG $\mathcal{G}_i$ constructed by using design and assembly rules from Definitions 7 and 8, respectively. Dashed arrows show the flow of the inter-agent fusion messages.

In the example shown in Figure 2 variables M and T were instantiated. Consequently, in any nonempty intersection of two Markov boundaries there was a single non-instantiated variable. For example $B(\mathcal{V}_1') \cap B(\mathcal{V}_2') = \{CondC, M, T\}$.

Note that by using design and assembly rules from Definitions 7 and 8 we cluster factors from original multiply connected BNs.

C. Distributed Inference

The distributed inference in a fusion organization Ω_t is based on sharing results of belief propagation in local BNs. Agent A_i runs Algorithm 1 upon receiving from some other agent A_j distribution $P_{in}(X_j|\mathcal{E}_j^t)$ over a variable from a common separator $X_j \in S\langle \mathcal{V}_i, \mathcal{V}_j \rangle$. The algorithm outputs distributions $P_{out}(X_k|\mathcal{E}^t \backslash \mathcal{E}_k^t)$ over uninstantiated variables X_k in each separator except the separator $S\langle \mathcal{V}_i, \mathcal{V}_j \rangle$ containing variable X_j whose distribution $P_{in}(X_j|\mathcal{E}_j^t)$ was supplied by agent A_j. Agent A_i with n separators thus computes and communicates to other agents $n-1$ distributions $P_{out}(X_k|\mathcal{E}^t \backslash \mathcal{E}_k^t)$ for all uninstantiated variables $X_k \neq X_j$. $P_{out}(X_k|\mathcal{E}^t \backslash \mathcal{E}_k^t)$ is distribution over variable $X_k \in S\langle \mathcal{V}_i, \mathcal{V}_k \rangle$ shared with agent A_k.

Algorithm 1: Local fusion algorithm

procedure : `LocalFusion(`$P_{in}(X_j|\mathcal{E}_j^t)$`)`

initialization: Create a list $\mathcal{I}$ of uniform distributions $\mathcal{I} = \{P_{in}(X_1|\mathcal{E}_j = \emptyset), \ldots, P_{in}(X_n|\mathcal{E}_j = \emptyset)\}$ for non-instantiated variables in each separator. Instantiate all context variables $X_p \in \mathcal{B}_i$ (i.e. set $P(X_p)$ to corresponding point mass distributions).

1 Replace the distribution $P_{in}(X_j|\mathcal{E}_j)$ in list $\mathcal{I}$ with $P_{in}(X_j|\mathcal{E}_j^t)$ received from agent A_j: $P_{in}(X_j|\mathcal{E}_j) \leftarrow P_{in}(X_j|\mathcal{E}_j^t)$;

2 For the uninstantiated variable X_k in each separator $S\langle\mathcal{V}_i, \mathcal{V}_k\rangle$ except $S\langle\mathcal{V}_i, \mathcal{V}_j\rangle$ compute:

$$P_{out}(X_k|\mathcal{E}^t \backslash \mathcal{E}_k^t) = \alpha \sum_{\mathcal{V}_i \backslash X_k} P(\mathcal{V}_i) \prod_{P_{in}(X_m|\mathcal{E}_m) \in \mathcal{I} \backslash \{P_{in}(X_k|\mathcal{E}_k)\}} P_{in}(X_m|\mathcal{E}_m) \prod_{X_p \in \mathcal{B}_i} P(X_p),$$

where α is a normalizing constant;

3 For each uninstantiated separator variable X_k except X_j communicate $P_{out}(X_k|\mathcal{E}^t \backslash \mathcal{E}_k^t)$ to the agents whose local models contain X_k;

4 For any variable $X_k \in \mathcal{V}_i$ we can compute posterior distribution given the entire current evidence $\mathcal{E}^t$:

$$P^t(X_k|\mathcal{E}^t) = \alpha \sum_{\mathcal{V}_i \backslash X_k} P(\mathcal{V}_i) \prod_{P_{in}(X_m|\mathcal{E}_m) \in \mathcal{I}} P_{in}(X_m|\mathcal{E}_m) \prod_{X_p \in \mathcal{B}_i} P(X_p),$$

Given rule 2 from Definition 8, Ω_t corresponds to a tree, where agents are represented by nodes while separators correspond to links. If we see agent A_i as a root of this tree, then we can identify a branch connected to A_i via agent A_k. In this context, $\mathcal{E}^t \backslash \mathcal{E}_k^t$ denotes evidence which has been injected into the entire system of agents that are in the fusion organization Ω_t except the branch connected to A_i via agent A_k up to the time t. Similarly, $\mathcal{E}_j^t$ denotes evidence that has been injected into the portion of Ω_t corresponding to the branch connected to A_i via agent A_j.

Note that the local computation of $P_{out}(X_k|\mathcal{E}^t \backslash \mathcal{E}_k^t)$ and $P^t(X_k|\mathcal{E}^t)$ can be based on any standard approach to belief propagation in arbitrarily complex local BNs, such as Junction tree propagation [7]. Since a local BN of an agent does not change, the local Junction tree in each agent can be computed prior to the operation. Also, the presented algorithm guarantees that every variable in any local BN is updated as a new piece of evidence is entered into an arbitrary local BN.

Proposition 2: A fusion organization of agents Ω_t implements correct belief propagation if (i) each agent runs algorithm 1, (ii) local BNs are obtained by using design rules from Definition 7 and (iii) the assembly of a fusion organization is based on rules from Definition 8. See the proof in [9].

In such a system, we can implement propagation which is equivalent to the *two-phase propagation*; i.e. marginal probability of every variable correctly reflects all priors and the entire evidence injected into different parts of Ω_t.

Note, in contrast to other approaches to distributed belief propagation in BNs [5], [7], the presented method does not require any compilation of Junction trees spanning over several agents. Instead, the presented approach makes use of local BNs directly. Due to the induced conditional independencies the partial results from the local fusion processes can be shared without any synchronization of different fusion processes.

IV. Discussion

We showed that causal Bayesian networks (BN) facilitate design of efficient and robust distributed fusion systems, since (i) they support compact representation of complex uncertain relations between heterogeneous variables and (ii) allow efficient inference (i.e. computation of probability distributions). BNs explicitly capture conditional independence, which reflects locality of causal relations. This allows very efficient modeling and reasoning about domains where each event is significantly influenced only by a small fraction of other events. For example, in monitoring systems hardware components or reports from one sensor do not influence hardware or reports from another sensor.

We can exploit locality of causal relations for efficient distribution of models and inference processes. By using the concept of Markov boundaries, we introduced design and assembly rules which can be used for building flexible distributed fusion systems. The design rules prescribe how local BNs of basic fusion modules are obtained through partitioning of monolithic BNs. The assembly rules, on the other hand, prescribe how the basic fusion modules must be organized and how the dependencies between the modules are reduced through instantiation of certain variables. Intersections of Markov boundaries provide guidance for the choice of variables that should be instantiated. The resulting distributed fusion systems have the following properties: (i) they support exact belief propagation without any computation of junction trees spanning several agents; each agent runs an arbitrary algorithm for exact belief propagation in arbitrarily complex local BNs. Locally, agents can use Junction trees, which are compiled prior to the operation; (ii) they do not require any centralized control of fusion or organization processes (see [9]).

Such fusion systems support exact belief propagation even in settings where constellations of information sources are not known prior to the operation and can change at runtime.

The presented approach is applicable to arbitrarily complex BNs. However, the approach is efficient if the intersections of Markov boundaries of different local BNs are relatively small. Fortunately, this is often the case with the causal models of monitoring processes. Such BNs typically consist of several network fragments which are conditionally independent given small sets of variables.

A relevant question is, whether the assumption about deterministic observations of context variables is realistic. Often these variables represent phenomena which can easily be measured with cheap and reliable sensors (e.g. temperature, humidity, wind speed, etc.). In addition, context variables serve as "global" variables used by many agents. Consequently, we might introduce a sensor of very high quality, whose measurements are communicated to several fusion agents with the corresponding context variable; i.e. in such cases investment in expensive equipment pays off.

Also, it can be very difficult to obtain CPT parameters precisely describing the true distributions over the combinations of the propensity states and states of the related phenomena. However, with the help of the recently introduced theory of Inference Meta Models [11], we can show that certain types of BN topologies in combination with instantiation of variables support very accurate estimation with asymptotic properties, even if we use parameters that deviate from the true distributions significantly. This is the case if (i) the domain models feature many conditionally independent fragments and (ii) the modeling parameters correctly capture very simple greater-than/smaller-than relations between the probabilities in the true distributions[7]. We can assume that such relations can easily be identified by experts or extracted from relatively small data sets with the help of machine learning techniques. Note that models of monitoring processes often feature inherently robust topologies, which is a consequence of locality of causal relations.

The introduced design principles were used in *Distributed Perception Networks (DPN)* [8], [9], a MAS framework supporting design of complex agent based fusion systems. DPN are in essence distributed self organizing classifiers, which consist of agents capturing heterogeneous domain expertise. Agents wrap information sources and provide uniform communication and fusion protocols. In contrast to other approaches to distributed inference in BNs, the DPN framework supports theoretically correct fusion which does not require any compilation of secondary fusion structures (e.g. junction trees and [7]) that span several modules. As new information sources wrapped by DPN agents enter the scene, the domain models of fusion systems are adapted on the fly, without any centralized control. DPN agents wrapping mobile sensor suites supply local BNs, arbitrarily sophisticated sensor models, which are plugged into the overall system as the agents join the fusion organization. In other words, each agent contributes a local model which relates agent's observations with the rest of the distributed causal model.

The presented approach to distributed fusion is complementary to other well known approaches to inference with distributed Bayesian networks, such as [5], [7]. In contrast to our method, these approaches cannot efficiently cope with domains where information source constellations are not known prior to the operation and can change at runtime. MSBN approach [7] is a sophisticated computational framework where agents collaboratively compile a secondary probabilistic structure spanning several agents. This, however, requires expensive processing and massive messaging which, in turn can be impractical in domains where constellations of information sources change rapidly. In such domains we might never end up with a viable global structure, since the time to obtain a secondary structure might be too long. The MSBN principles have been used as a basis of the approach in [3], which makes use of Markov boundaries in order to reduce the complexity of inference processes in MSBNs. In this approach, agents collaborate to find observable Markov boundaries. However, in contrast to our approach, it does not use Markov boundaries in the design phase (distribution) and it does not avoid the compilation of secondary structures spanning several agents. Another well known approach to distributed inference in Bayesian networks is robust message passing [5], which requires that the full model of the problem domain is known prior to the operation. Again, this requirement is a serious drawback in domains that are targeted by the DPN approach. There exist also approximate inference methods. Particularly interesting is loopy belief propagation [12], which avoids compilation of Junction trees or reduction of independencies through instantiation of variables. However, loopy belief propagation does not guarantee a good convergence and it requires massive messaging.

[7]For example, assume the true conditional probabilities over binary variables E and C: $P(e|c) = 0.7$, $P(\bar{e}|c) = 0.3$, $P(e|\bar{c}) = 0.4$ and $P(\bar{e}|\bar{c}) = 0.6$. We say that a conditional probability table correctly captures relations between these probabilities if its parameters (i.e. conditional probabilities $\hat{P}(E|C)$) satisfy very simple relations: $\hat{P}(e|c) > 0.5$ and $\hat{P}(e|\bar{c}) < 0.5$.

REFERENCES

[1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[2] Kschischang, Frey, and Loeliger, "Factor graphs and the sum-product algorithm," *IEEETIT: IEEE Transactions on Information Theory*, vol. 47, 2001.

[3] X. An, Y. Xiang, and N. Cercone, "Revising markov boundary for multiagent probabilistic inference," in *Procs. IEEE/WIC/ACM Inter. Conf. on Intelligent Agent Technology*, 2004, pp. 113–119.

[4] R. Neapolitan, *Learning Bayesian Networks*. Prentice Hall, 2003.

[5] M. Paskin and C. Guestrin, "Robust probabilistic inference in distributed systems," in *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI-04)*. Banff, Canada: AUAI Press, 2004, pp. 436–445.

[6] N. Jennings, K. Sycara, and M. Wooldridge, "A roadmap of agent research and development," in *Autonomous Agents and Multi-Agent Systems*, July 1998, vol. 1, no. 1, pp. 7 – 38.

[7] Y. Xiang, *Probabilistic Reasoning in Multiagent Systems: A Graphical Models Approach*. Cambridge University Press, 2002.

[8] G. Pavlin, P. de Oude, and J. Nunnink, "A mas approach to fusion of heterogeneous information," in *Proceedings of the The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, 2005, pp. 802–804.

[9] G. Pavlin, P. de Oude, M. Maris, J. Nunnink, and T. Hood, "A distributed approach to information fusion based on causal probabilistic models," Technical Report IAS-UVA-07-03, Informatics Institute, University of Amsterdam, The Netherlands, Tech. Rep., 2007.

[10] P. de Oude, B. Ottens, and G. Pavlin, "Information fusion in distributed probabilistic networks," in *Artificial Intelligence and Applications*, Innsbruck, Austria, 2005, pp. 195–201.

[11] G. Pavlin and J. Nunnink, "Inference meta models: Towards robust information fusion with bayesian networks," in *The Ninth International Conference on Information Fusion*, Florence, Italy, 2006.

[12] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *In Proceedings of Uncertainty in AI*, 1999, pp. 467–475.