A New Framework for Inference in Distributed Bayesian Networks for Multi-Agent Sensor Interpretation*

Norman Carver
Computer Science Department
Southern Illinois University Carbondale
Carbondale, IL, USA
carver@cs.siu.edu

Abstract

Multi-agent systems (MAS) are groups of interacting intelligent software agents. An important application is *sensor interpretation* (SI) in sensor networks. SI domains are frequently modeled with *Bayesian networks* (BNs), and distributed versions of these problems can be modeled with *distributed Bayesian networks* (DBNs). The *multiply sectioned Bayesian network* (MSBN) framework is the most studied approach for inference in DBNs, in an MAS setting. However, we do not believe the MSBN framework is well suited for large-scale MAS-based SI. This paper describes an alternative framework for inference in DBNs that we have developed to support efficient, approximate MAS-based SI. Compared to the MSBN approach, our approach supports more autonomous and asynchronous agents, and more focused, situation-specific communication patterns. Our analyses show that this framework can be used to produce acceptable interpretations at substantially lower cost than the MSBN.

1 Introduction

Multi-agent systems (MAS) are groups of intelligent software agents interacting to achieve one or more objectives. An important application for MAS is sensor interpretation (SI) in sensor networks. The major advances that have occurred in sensor and wireless technology have made it practical to deploy large networks of low-cost sensors. Unfortunately, there have not been comparable advances in the software to control such networks and process the vast amounts of data they can produce. MAS approaches offer opportunities for dealing with large-scale sensor networks due to factors such as distributed control and computation. However, research is still required to understand how to build effective large-scale MAS.

SI domains can frequently be modeled with *Bayesian networks* (BNs) and distributed versions of these problems can be modeled with *distributed Bayesian networks* (DBNs). The *multiply sectioned Bayesian network* (MSBN) framework, e.g., [3, 4, 5], is the most studied approach for using DBNs in an MAS setting. However, we do not believe the MSBN framework is well suited for MAS-based SI in large-scale sensor networks.

This paper introduces a new framework for probabilistic inference in DBNs, which has been designed to support flexible and efficient approximate MAS-based SI. Compared to the MSBN approach, our framework supports more autonomy and parallelism among the agents, and more dynamic, situation-specific communication patterns. To compare the performance of the two approaches, we will present some analyses of the time required to produce interpretations. The analyses show that in at least some sensor network domains, our framework can be used to produce acceptably high quality solutions at substantially lower cost.

The next section provides basic background knowledge on sensor interpretation, Bayesian network algorithms, and the MSBN framework. The following two sections introduce our alternative framework and analyze the relative costs and performance of the two approaches. The paper concludes with a summary and future research section.

2 Background: SI, BNs, and MSBNs

The goal of sensor interpretation is to identify the set of *events* in the environment that are responsible for producing the sensor data. Such a set is termed an *interpretation*. There will typically be multiple possible interpretations for most data sets, so an SI system must determine the "best" interpretation. One of the most commonly used probabilistic standards is the

*This material is based upon work supported by the National Science Foundation under Grant No. IIS-0414945.

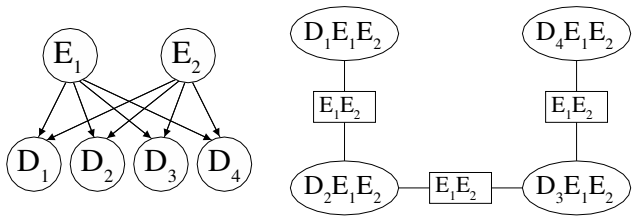

Figure 1: A simple SI BN and its (optimal) join tree.

maximum a posteriori interpretation or MAPI.[1] The MAPI is the *optimal* solution when event priors are known, and would be considered an "exact solution." While an SI system would ideally always compute the MAPI of all of the globally available sensor data, computing the MAPI is *NP-hard* in general, and is virtually always impractical in real-world sensor networks.

A typical MAS-based approach to SI in a sensor network will have the sensors partitioned among the agents, and each agent would be charged with identifying whether a certain subset of the events had occurred. Each agent ends up with direct access to only a subset of the sensor data (its *local data*), so agents generally have to communicate to solve their subproblem(s). While communication is necessary, it will always be overhead as relative to processing data to determine the best interpretation, so it must be limited. This is particularly true since communication is inherently much slower than computation and can consume considerable resources (e.g., network bandwidth or battery energy in wireless sensor networks).

The most popular approach for doing probabilistic inference these days is with Bayesian networks. Figure 1 shows a very simple BN that has the basic general form necessary to define an SI domain. The network has two levels: one for the events (E_1 and E_2) and one for the data (D_1 through D_4).[2] The BN is *multiply connected* (i.e., contains undirected loops), as are the BNs for virtually all SI domains. The standard technique for doing exact inference in multiply connected BNs is to convert the BN to a *join tree*, and operate on that structure. The standard join tree version of this BN is also shown in Figure 1.

We will give only the briefest overview of the join tree approach. An excellent introduction can be found in [1]. Each node in a join tree is a *cluster*: a nonempty set of (random) variables (i.e., events, data, etc.). Each edge is labeled with a *sepset*: the intersection of the variables in the two clusters connected by the edge. In the join tree in Figure 1, clusters are ovals and sepsets are rectangles. Every cluster and every sepset has an associated *belief potential*, which is effectively an unnormalized probability distribution over the combinations of values of its variables.

Inference in a join tree involves a *global propagation procedure* that consists of a sequence of *message passes* that distribute the probability information among the clusters and sepsets. A message pass from Cluster A to Cluster B via their connecting sepset involves marginalizing Cluster A's potential to generate a new sepset potential, dividing out the old sepset potential, and multiplying Cluster B's potential by this change potential. When new evidence is received, it must be entered into the join tree and the global propagation procedure rerun. Each cluster (or sepset) then contains a potential that effectively represents the conditional probability of its variables.

In the *multiply sectioned Bayesian network* (MSBN) framework, a (large) BN is broken up into subnetworks such that the resulting structure is effectively a join tree, with each subnetwork as a cluster and the subnetwork connections as sepsets. This structure is referred to as the *communication graph* of the MSBN [5]. The messages that would be communicated between the subnetworks/agents in the MSBN are basically the same as the sepset messages that would be passed in a join tree. An example MSBN is shown in Figure 2. Here the BN of Figure 1 has been divided into two sub-BNs, as when each agent has direct access to half the sensors. Each agent's subproblem could be to identify the correct value for one of the events. The agents can pool the effects of their local data by communicating messages based on their E_1-E_2 potentials.

The MSBN approach was originally developed for large centralized BNs. Its extension to MAS does not provide any adaptations for dealing with the differences between centralized and multi-agent frameworks. We see at least six significant drawbacks in applying the MSBN framework to MAS-based SI: (1) agent autonomy is restricted; (2) agents are idled during significant portions of the global propagation process; (3) global consistency is forced on the system if any data evidence is to be shared; (4) few techniques for approximate SI are supported; (5) agent communication patterns are constrained; and (6) it is unclear how the MAS determines when to initiate the global propagation process.

Though the agents in an MAS-based sensor network are inherently cooperative, they are also generally intended to have a degree of autonomy and asynchrony. The MSBN procedures represent an "all or nothing" approach to MAS communication. The only way to communicate data evidence among agents is to

[1] MAPI(D)= argmax($P(I_i \mid D)$), where I_i represents a possible interpretation and D represents the observed data.

[2] Note that introducing nodes (random variables) intermediate between the event and data levels does not add to the power to *model* domains, though when representing appropriate abstractions, such nodes can reduce computational costs.

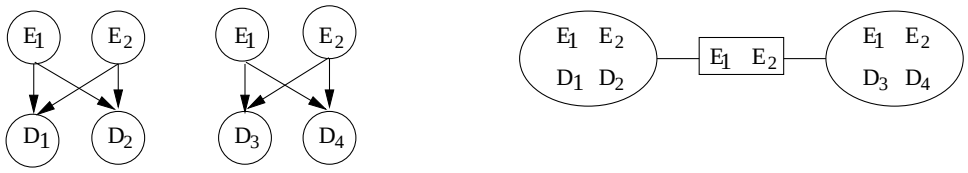

Figure 2: The subnetworks of an MSBN and the resulting communication graph.

involve the entire network of agents in the global propagation process, communicating the effects of *all* data processed by *all* agents to *all* agents. Exploiting the parallelism inherent in an MAS is typically crucial for SI, but in the MSBN approach, agents cannot continue to process local sensor data during portions of the global propagation process (so called "off-line time" [4]). The global propagation process results in every agent having the same probability distribution knowledge about the set of event variables it is working with, but this duplication largely defeats the purpose of having different agents responsible for interpreting different events. Because the MSBN requires that the communication graph have a tree structure, some agents will be able to communicate only by passing messages via another agent, delaying the receipt of potentially critical evidence and using additional computational resources. Overall, the MSBN approach will often require substantially more resources than necessary, particularly where various SI approximations are effective. Finally, the MSBN work has not addressed the question of how an MAS determines when the global propagation process should be initiated.

3 An Alternative Framework for MAS-based SI Inference in DBNs

Because of the limitations of the MSBN approach, we have developed an alternative framework for probabilistic inference in DBNs, specialized for MAS-based SI. This framework has been designed specifically to support more autonomy and parallelism among the agents, more flexible and dynamic communication patterns, and more approximation techniques. The two key approximations that we wish to support are: (1) having each event value determined by a single agent, independently of the other events; and (2) allowing agents to base their solutions on different subsets of the globally available data.

The first key difference between our DBN approach and the MSBN approach is that in our approach each agent maintains its own "virtual sepset" data structure for *each* connection to another agent (so two sepsets per link). The MSBN uses a single, shared sepset for each linkage.[3] Because we want to allow agents to operate autonomously, we cannot have an agent assume that just because he sent evidence to a second agent, the second agent has processed this evidence (or ever will). Only the receiving agent can reliably determine what communicated evidence it has processed and thus must be "divided out" from any evidence that it sends back to the other agent. Maintaining two sepsets allows each of two directly connected agents to make independent decisions about what evidence sent by the other will be processed and integrated into their own model, and when.

We now describe how the two virtual sepsets per connection are utilized. Each agent's events potential and all of its virtual sepsets are initialized with their joint events priors. When an agent "processes" local data, its events potential will be updated as normal in a BN. To communicate data evidence to another agent, the originating agent sends the other agent a message containing an *events likelihood vector* (ELV), which is an appropriately modified version of its events potential. We impose two constraints on these ELVs: (1) they must not reflect data evidence that originated with the target agent (to avoid "double counting" this evidence); and (2) successive messages must reflect incrementally more local data (not just additional data). These constraints allow the agents to operate independently and asynchronously, and yet still pool their data evidence. An agent's virtual sepset for a connection is used in two ways. First, when an agent is going to send an ELV to the other agent, it uses its sepset to remove the effect of any previously communicated *and processed* evidence from the other agent (by "dividing it out"). Second, when the agent decides to process an ELV from the other agent, it uses its sepset to remove the effect of any evidence it has previously processed from the other agent.

The second key difference between our DBN approach and the MSBN approach is that we do not require that agents be linked in only a tree structure. Instead, we allow any pair of agents to be linked and directly communicate, and in particular, we allow the

[3] While [4] makes it clear the MSBN is to have a single sepset associated with each linkage, nowhere could we find an explicit statement of which agent the sepset data structure is to be stored with nor which agent is to do the relevant computations.

network of agents to be fully connected. This potentially introduces one or more loops into the agent communication graph, resulting in multiple propagation paths for the same data evidence, and the possibility for data evidence to be "counted" multiple times. For example, suppose we have a fully connected network of three agents, A_1, A_2, and A_3. Now A_1 communicates an ELV with its local data to A_2, A_2 then communicates an ELV representing both its and A_1's data to A_3, and A_3 ultimately communicates an ELV representing all agents' local data to A_1. At this point, A_1's interpretations potential could end up "double counting" the effects of its own local data and so would represent an invalid conditional probability. [5] contains a discussion of the effect of loops in the MSBN communication graph. It points out that what it terms "degenerate loops" (loops where the sepsets all share some variables) can be dealt with by arbitrarily breaking the loop. The loops that would occur in SI DBNs would typically be of this type. For example, we could arbitrarily decide to break the above loop between agents A_1 and A_3. Now, though, if A_1 needs data from A_3, A_3 will have to prepare and communicate an ELV to A_2, which will then have to prepare and communicate an ELV to A_1. Not only will this take twice as long to get information to A_1 as direct A_3 to A_1 communication, A_2 is forced to do computation that may be of little/no value to its subproblem. While randomly breaking a loop prior to doing a global propagation in an MSBN is appropriate, prohibiting all communications between certain pairs of agents is not ideal in general.

Instead, we have developed a number of approaches that allow us to limit what data evidence is communicated between certain agents, without having to prohibit all communications between these agents. For example, we can eliminate the "loop problem" in the above three agent system by deciding at design time that each of these agents should communicate only their own local data evidence to another agent; they will never propagate data evidence that originated with another agent. How can this be done, though, since once an agent processes an ELV from another agent, its events potential reflects data evidence other than its own? It turns out that our dual virtual sepset mechanism makes this possible. Suppose A_2 has processed an ELV from A_1, so A_2's events potential now reflects both its local data and the local data of A_1. Instead of A_2 simply using its sepset associated with its A_3 linkage to prepare the ELV for A_3, it can also use its sepset associated with its A_1 linkage to remove the effects of A_1's data evidence. This approach can obviously be generalized, allowing an agent to send a likelihood vector message that represents its own local data as well as evidence from any desired subset of the agents it has received evidence from. The cost for this capability is the additional sepset computations (divisions) that must be done to remove evidence that should not be propagated further. If there are agents that receive evidence from a large number of agents, but must then send only their own local data evidence, the scheme we just proposed would become inefficient. The alternative that we would then propose is for such an agent to maintain a second joint events potential that will always reflect only its local data, which it could use in constructing ELV messages. Due to space limitations, we will not go into the details of this scheme here.

While statically determined communication patterns provide more flexibility than the MSBN tree approach, they still limit the flexibility of the MAS to respond to the situation it finds as data is processed. Ideally, we would like agents to be able to dynamically determine the most appropriate communication patterns. The approach we have just outlined allows agents to limit the evidence they propagate to other agents, but how can an agent determine what evidence needs to be eliminated from a message. To do this, each agent needs to understand what data has been integrated into its current events potential as well as what data has been incorporated into the target agent's events potential. We propose that this be accomplished by having agents maintain a global dataset "bit vector" along with their joint events potential and along with every virtual sepset, and that agents send this information in their evidence messages along with the ELVs.. The dataset bit vector would identify what data is contained in a potential/likelihood vector, and allow agents to avoid double counting evidence. If an agent still receives an ELV that cannot be integrated because it contains data evidence it has already processed, it can avoid invalidating its potential, and send a request to the source agent to eliminate the offending evidence (as we discussed above) and resend the evidence.

4 Performance Analysis

Taken together, the elements of our approach to DBN inference provide a much more flexible framework for MAS-based SI. However, there are costs to what we propose—just as there are costs to the MSBN's global propagation procedure. In this section, we will present some analyses of the time requirements and solution quality performance of the MSBN approach and approaches based on our framework. We will use the following notation:

- e – number of events;
- a – number of agents;

- C_* – time per floating point multiplication;
- C_f – fixed time per communication;
- C_v – time to transmit one floating point number
- $P = 2^e \cdot C_*$ – time to process one ELV;
- $S = 2^e \cdot C_v$ – time to transmit one ELV message;
- $M = C_f + S$ – total time to send one ELV message;

Let us first consider how to derive parametrized expressions for the time cost of the MSBN global propagation procedure. The appropriate expression will depend on the topology of the communication graph and on assumptions about the ability of multiple agents to communicate in parallel with a single agent. The MSBN requires that the communication graph always be a tree structure. Suppose this is a "star structure," where all agents $(a-1)$ are directly linked to a single central/root agent, and let us assume here that all agents can communicate in parallel to the central agent. Each agent must first spend P time doing computation to prepare its ELV, but can do this in parallel. They then send a message containing the ELV to the central agent, again in parallel, in time M. The central agent receives $a-1$ messages at essentially the same time, and must then process all of them (one-by-one) to update its events potential, requiring a time of $(a-1)P$. The MSBN global propagation procedure then propagates the combined results back to all the agents, and this takes exactly the same amount of time as the inward propagation and P time for each receiving agent to integrate the message. Thus, the total time for the global propagation process is: $2M + 2aP$.

We will now consider a more general tree structure, where the uniform depth of the tree is d, the uniform branching factor is b, so $a = \frac{b^{d+1}-1}{b-1}$. Let us again assume all agents communicating with a single agent can do so in parallel. Under these assumptions, we can express the time required for the global propagation procedure as: $2dM + 2d(b+1)P$. Comparing this to "star configuration," we see that the time due to local data processing has been reduced ($(a-1)P$ vs. $d(b+1)P$, where generally $db << a$) since the different levels of agents are merging results in parallel. However, this comes at the cost of increased communication time (M vs. dM). In most reasonable sensor networks, M is greater than P (often by orders of magnitude), so this is rarely going to be a good trade-off.

The other time-related aspect of the MSBN global propagation procedure is the "offline time" for the agents, when they sit idle, unable to process any local data. With the star configuration, all agents except the central agent are offline during the entire propagation process. The central agent is offline for less time: $2(a-1)P$. For the more general tree structure, the offline time for the central agent is unchanged, but the offline for other agents will depend on their distance (depth) from the central/root agent. The average time for an agent at depth d is: $2dM + 2((d-1)(b+1) + b/2 + 1)P$.

Consider now a situation in which the agents receive and process data incrementally, and all agents need to be updated after each batch of data has been processed. Suppose each agent receives its m pieces of data in two batches of size $m/2$. The MSBN approaches will entail significant offline time for all the agents during the global propagation process. Effectively, the agents will process half their data, do the global propagation, then process the other half, then do the global propagation again. There will be no useful parallelism between the data processing and the propagation, so the total time to produce a solution will be the time to process the data plus the time for two global propagations. For the star configuration this means a time of: $\frac{m}{2}P + 2(M + aP) + \frac{m}{2}P + 2(M + aP)$ or $4M + (m + 4a)P$. For the more general tree structure, it will be: $4dM + (m + 4d(b+1))P$.

Let us now consider how to derive expressions for the time cost of our approach. Assume a fully connected communication graph and parallel communication capabilities. As a base comparison, consider updating all agents with all data evidence, though this will not be an efficient way to use our framework. Each agent must process its potential via its virtual sepset for each linked agent and then send the resulting ELV. The processing must be done sequentially, but since we are assume communication can occur in parallel (partially), the last agent receives the ELV at $(a-1)P + M$. Each agent must process the received message through its local sepset and use the result to update its potential, requiring $2(a-1)P$ time. However, because an agent can begin processing the first received messages while other messages are still being transmitted, the additional time required is only $((a-1)+1)P$ or aP. Thus, each agent has its potential updated after a time of: $M + (2a-1)P$. Even for this inappropriate use, our approach is less costly than the MSBN global propagation procedure here.

One of the key advantages of our approach over the MSBN approach is its ability to better support MAS parallelism. Our approach never has the agents sitting idle, as the MSBN approach does. This makes it more difficult to develop expressions for the time costs of incremental strategies, because agents are able to begin working on processing the second batch of data while the first communications are still being transmitted. If we again consider updating all agents with all data, but factor in the incremental processing of the m data per agent, we get the following time bound for our approach (assuming $\frac{m}{2}P > M > P$): $M + (m + 4(a-1) + 2)P$. This shows that our ap-

proach does more computation than the MSBN procedure, but the communication time plays a very small role due to parallelism. In fact, if we were continuously getting additional data, M would become irrelevant.

The above situation (all agents communicate all data to all agents) does not represent the intended usage of our approach. Our approach was designed to support limited, focused communication among agents, for approximate SI. If all agents need all the data to solve their subproblems, then the MSBN's global propagation procedure is reasonable. However, this will not be practical for most sensor networks. Let us now consider an approximate SI strategy that can take advantage of the features of our approach. One approximation that takes advantage of the agent structure is to have event values be identified separately rather than jointly as in the MAPI. For example, one agent will be responsible for determining whether E_1 is true or not, another agent whether E_2 is true or not, etc. We have previously considered this approximation [2], and have defined a measure of a domain's potential to support this approximation. The *event decomposability index* (EDI)of a domain is the probability that the conditional probability of each event considered separately (but using all data) is compatible with the global MAPI.[4] We have generated large numbers of random SI compatible BNs and analyzed their EDIs, and virtually all have indices over 95%. This suggests that this approximation is likely to be effective in many SI domains.

A key reason why pursuing events independently is of interest is that in examining hundreds of randomly generated SI-type BNs, we have found that individual event values can often be reliably identified with only a fraction of the global data, if the data is selected specifically for each event [2]. For example, we find that agents can often achieve a 99% probability of having the same value for an event as they would have from all the data, with access to only the most appropriate 10-20% of the data. This finding can be applied to the design of an effective and efficient MAS-based SI system by having agents work on one or more events, and having only select agents communicate their data to each other agent, based on that other agent's event(s) subproblem. Determining cost expressions for this type of strategy can be very difficult in general. However, if we assume that all agents both receive data from the fraction f of other agents and send data to the fraction f of other agents, the time for this strategy is: $M + (m + 4f(a-1) + 2)P$. If f is on the order of 0.2, this becomes approximately $M + (m + a)P$, which is much faster than the MSBN.

[4] If the conditional probability of the event given the data is greater than 0.5 then the event would be true, else it would be false. This would be compared to the T/F value of the event in the MAPI.

5 Conclusions

The multiply sectioned Bayesian network framework is the most studied approach to inference in distributed Bayesian networks, and it has been championed as a method for sensor interpretation with multiagent systems. However, we do not believe it is a practical approach for SI in large-scale sensor networks. Instead, successful approaches will need to be able to support agent autonomy and parallelism, limited and dynamic communication, and various approximation methods. In this paper, we have introduced such a framework designed specifically for MAS-based SI. While our analyses showed potential costs advantages of our approach, this paper does not showcase the ability of our approach to support highly dynamic MAS strategies. Future work will focus on identifying strategies that use our inference mechanisms to produce appropriate quality solutions with reasonable costs. We are particularly interested in considering energy as a cost, since this is critical for many wireless sensor networks, and makes it even more important that communication be limited.

References

[1] Cecil Huang and Adnan Darwiche, "Inference in Belief Networks: A Procedural Guide," *International Journal of Approximate Reasoning*, vol. 11, 1–158, 1994.

[2] Muralidar Chakravarthi, *Domain Decomposability and Efficient Strategies for MAS-Based Sensor Interpretation*, MS Thesis, Computer Science Department, Southern Illinois University Carbondale, 2000.

[3] Yang Xiang, David Poole, and Michael Beddoes, "Multiply Sectioned Bayesian Networks and Junction Forests for Large Knowledge-Based Systems," *Computational Intelligence*, vol. 87, no. 1–2, 295–342, 1993.

[4] Yang Xiang, "A Probabilistic Framework for Multi-agent Distributed Interpretation and Optimization of Communication," *Artificial Intelligence*, vol. 9, no. 3, 171–220, 1996.

[5] Yang Xiang and Victor Lesser, "On the Role of Multiply Sectioned Bayesian Networks to Cooperative Multiagent Systems," *IEEE Transactions on Systems, Man, and Cybernetics-Part A*, vol. 33, no. 4, 489–501, 2003.