Efficient Approximate Inference in Distributed Bayesian Networks for MAS-based Sensor Interpretation

(Short Paper)

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

ABSTRACT

The multiply sectioned Bayesian network (MSBN) framework is the most studied approach for distributed Bayesian Network inference in an MAS setting. This paper describes a new framework that supports efficient approximate MAS-based sensor interpretation, more autonomy and asynchrony among the agents, and more focused, situation-specific communication patterns. Its use can lead to significant improvements in agent utilization and time-to-solution.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Design, Performance

Keywords

distributed problem solving, distributed Bayesian networks

1. INTRODUCTION

Distributed problem solving (DPS) is the subfield of multiagent systems that is concerned with solving large-scale, often inherently distributed problems, using systems of distributed intelligent agents. An important DPS application is *sensor interpretation* (SI) in sensor networks. SI domains can frequently be modeled with *Bayesian networks* (BNs) so data interpretation is basically via BN inference. Distributed, multi-agent SI can be modeled with *distributed Bayesian networks* (DBNs). In a DBN, sub-networks of the global BN are distributed to different agents, and some BN inferences will require communication among the agents.

The *multiply sectioned Bayesian network* (MSBN) framework, e.g., [4, 2, 3], 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. In large-scale applications, exact interpretation will be impractical, it will be critical to take advantage of the parallel computational capabilities of the agents, and agent coordination strategies may need to be flexible and able to dynamically adapt to the situation. Among the key problems we see with the MSBN are: it supports only exact inference, its global propagation procedure reduces agents' ability to work in parallel, agent autonomy and asynchrony is limited, and communication patterns among the agents are severely restricted.

This paper describes the elements of a new framework we have developed for inference in DBNs, which has been designed specifically to support efficient and flexible, 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 two approaches, we will present some mathematical analyses of the time required to produce interpretations using the MSBN and our framework. The analyses show that in at least some sensor network domains, it is likely that our framework could be used to produce acceptably high quality solutions at substantially lower cost than with the MSBN.

2. BACKGROUND

The goal of sensor interpretation is to identify the set of *events* in the environment that are responsible for producing the sensor data, termed an *interpretation* of the data. The most commonly used *exact* probabilistic standard for selecting the best interpretation is the *maximum a posteriori interpretation* or MAPI. Ideally, an SI system would report the MAPI of all of the globally available data as its solution (the "global MAPI"), but this is generally impractical, as computing the MAPI is *NP-hard*.

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 possible events had occurred (i.e., this would be the agent's *subproblem*). Each agent will end up with direct access to only a subset of the globally available sensor data, termed its *local data*. Agents will generally have to communicate and exchange results and/or data to solve their subproblems, because their subproblems will not be independent. Because communication will always be overhead relative to the system goal of processing data, it must be limited and focused (particularly since communication is inherently much slower than computation and can require considerable energy resources in wireless sensor networks).

Cite as: Efficient Approximate Inference in Distributed Bayesian Networks for MAS-based Sensor Interpretation (Short Paper), Norman Carver, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 1535-1538.

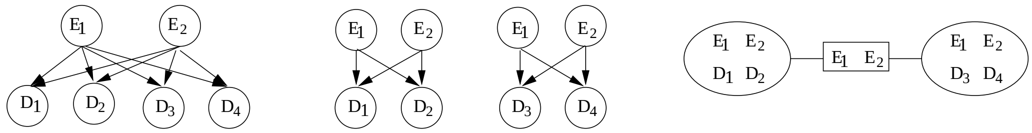

Figure 1: A simple SI BN, the MSBN subnetworks, and the MSBN communication graph.

Figure 1 shows a very simple BN that has the most 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). The BN is *multiply connected*, and a standard technique for doing exact inference in such networks is to convert the BN to a *join tree*, operate on that structure, and compute probabilities as needed. An excellent presentation can be found in [1]. Each node in a join tree is a *cluster*: a set of (random) variables. Each edge is labeled with a *sepset*: the intersection of the variables in the two connected clusters. Every cluster and every sepset has an associated *belief potential*, which is effectively an unnormalized probability distribution over its variables. Inference in a join tree involves a *global propagation procedure* that consists of a sequence of *message passes*. Upon completion, the join tree is said to be *consistent*, with the probability distribution having been propagated to *all* clusters and sepsets. A joint of interest can then be computed by *marginalizing* over *any* potential containing the joint 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, in the sense that each subnetwork can be viewed as a cluster, and the connections between these subnetwork clusters can be viewed as sepsets. This structure is referred to as the *communication graph*, since each subnetwork would be distributed to a different agent and inference would require communication (in which case the sepsets are termed *linkages*). 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 with the subnetworks as the clusters.

An example MSBN is shown in Figure 1, based on the simple SI BN. Here, the original BN has been divided into two Bayesian subnetworks. This could correspond to a situation in which each agent has direct access to half the sensors. The agents can exchange and pool the effects of their data by processing their data to update their respective E_1-E_2 potentials, and then communicating messages based on the E_1-E_2 potentials. A BN cannot be arbitrarily decomposed into an MSBN: (1) the communication graph must have a *tree* structure; and (2) subnetworks must be *d-separated* by the sepsets/linkages.

The MSBN approach was originally developed for large centralized BNs. Its extension to MAS provides few 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) global consistency is forced on the system if any data evidence is to be shared; (2) agents are idled during significant portions of the global propagation process; (3) agent autonomy and asynchrony are reduced; (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.

Communication takes places only via a global propagation procedure analogous to that in a join tree. This involves the entire network of agents communicating the effects of *all* data processed by *all* agents to *all* agents. To a large extent this offsets the advantages of having different agents responsible for different events. Global propagation severely limits the ability to exploit the parallelism inherent in an MAS, as agents must sit idle during period of the global propagation process ([2] terms this "off-line time"). Agents have little autonomy. Once the propagation procedure begins, an agent's activities will be outside of its control; they must must process all messages they receive, whether the contents are useful for their subproblem or not. Because the MSBN requires that the communication graph have a tree structure, some agents will be able to communicate only by passing messages via other agents, delaying the receipt of potentially critical evidence and using additional computational resources (in the intermediate agent(s)).

3. A NEW FRAMEWORK

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 to support better utilization of the concurrent computational capabilities of the agents, more flexible and focused communication patterns, increased agent autonomy and asynchrony, and a variety of approximation techniques for SI. The two main approximations that we wish to support are: (1) having each event value determined by a single agent; and (2) allowing agents to base their solutions on different subsets of the data.

The first element of our approach involves the structure of each agent's local BN. For SI there is typically no need for global propagation procedures that produce correct probabilities throughout the BN. Instead, we care about producing he correct potential only for a single cluster containing all the events of interest to the agent. It can be shown then that a using "central cluster" join tree model is more efficient than any other join tree topology. A system of such agents are linked into a distributed BN by passing appropriate messages between agents' events central clusters, via "virtual sepsets." To communicate results from its processing to another agent, an agent sends the other agent a message containing what we term an *events likelihood vector* (ELV), which is derived from its central cluster events potential. When an agent processes local data or a received ELV, its central cluster events potential will be updated so that it reflects a revised conditional probability distribution.

The first key difference between our approach and the MSBN approach is that each of our agents maintains its own "virtual sepset" data structure for *each* connection to another agent (so there are two sepsets associated with each link). The MSBN uses a single, shared sepset for each linkage. Not only do we want to support agents that can maintain inconsistent interpretation potentials (i.e., based on different sets of data), we want agents to be able to operate autonomously and asynchronously. Just because one agent has sent a message to a second agent, the second agent should not be forced to process it. Only the receiving agent can reliably determine what communicated evidence it has processed and thus must be "divided out" from any messages 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, and when.

Communicating sensor data evidence to another agent means sending a message with an events likelihood vector that is an appropriately modified version of the joint events potential in its central cluster. We impose two constraints on the ELVs that are communicated: (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 both previous and additional local data (not just additional data). These constraints allow the agents to make autonomous and asynchronous decisions about when to send and process messages, while still allowing them to pool their processed data. In our basic communication scheme, an agent's virtual sepset for a connection is used in two ways: (1) When an agent is going to send an ELV to another agent, it uses *its* sepset associated with the link to that other agent to remove the effects of any previously communicated *and processed* evidence from the other agent (by "dividing it out"). (2) When the agent decides to process an ELV from the other agent, it first uses *its* sepset associated with the link to the other agent to remove the effect of any evidence it has previously processed from the other agent, and then updates its potential to reflect all processed evidence from the other agent.

The second key difference between our approach and the MSBN approach is that we do not require that agents be linked in only a tree structure (for communication). Instead, we allow any pair of agents to be linked and directly communicate. 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. [3] 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, and the loops that would occur in SI DBNs would often be of this type. However, while randomly breaking a loop is appropriate for the MSBN's global propagation procedure, prohibiting all communications between certain pairs of agents is not ideal as it slows communication between some agents and forces intermediate agents to do work that may be of no value in solving their own subproblems.

Instead, we have developed two variations on our basic communication scheme that allow us to limit what data evidence is communicated between certain agents, without having to prohibit all communications between these agents. In the first variation, we take advantage of our dual sepset per link mechanism to allow an agent to limit the evidence that it passes on. Just as an agent can use its linkage virtual sepset to avoid sending evidence back to the agent it received it from, the fact that it has comparable information for each link to another agent means that it can divide out (remove) evidence from any combination of connected agents before it sends out an ELV. This effectively allows for arbitrary communication patterns to be statically determined and supported by the agents. The main cost for this capability is the additional sepset computations (divisions) that must be done to remove evidence (it is actually a bit more complicated than this due to the priors). 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 could become inefficient. The alternative scheme 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. When a batch of local data is processed, this local events potential would be updated first, and then it would periodically be used to update the central cluster (which would as before reflect both local data and data evidence received and processed from other agents). An agent can then decide to use whichever of the its two events potentials are appropriate to construct an ELV message for a particular agent, depending on whether it wants to communicate only its local data evidence to that agent or whether it wants to pass on data from one or more agents in addition.

The schemes we have just outlined allow agents to limit what evidence they pass on to other agents, allowing an MAS designer more flexibility in specifying communication patterns than the MSBN's tree topology. However, the communication patterns must be statically determined. Ideally, we would like agents to be able to dynamically adjust their communication patterns. To do this, an agent would have to be able to determine what evidence needed to be eliminated from each ELV, and have the ability to eliminate it. This requires that an agent 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 an ELV, and so 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 and resend the evidence.

4. PERFORMANCE ANALYSIS

In this section, we will present some analyses that compare the time-to-solution performance when using the MSBN framework and approaches based on our framework, for exact and approximate SI. This will allow us to assess the trade-offs. We will use the following notation in the analyses:

- 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;
- $M = C_f + 2^e \cdot C_v$ – total time to send one ELV message;

Consider how to derive parametrized expressions for the time cost of the MSBN global propagation procedure. The cost will depend on the topology of the communication graph and on assumptions about the ability of agents to communicate "in parallel." The MSBN requires that the communication graph 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 time P doing computation to prepare its ELV, but can do this in parallel. They then send a message containing the ELV to the central agent, in parallel, in time M. The central agent receives $a-1$ messages at (essentially) the same time, and must then process them to update its events potential, requiring time $(a-1)P$. 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 time P for each receiving agent to integrate the message. Thus, the total time is: $2M + 2aP$.

Now consider a more general tree structure, where the uniform depth of the tree is d, the uniform branching factor is b, so $a = \frac{b^{d+1}-1}{b-1}$. 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 $db << a$). However, this comes at the cost of increased communication time (M vs. dM). In most sensor networks, M is greater than P (often by orders of magnitude), so this is rarely going to be an effective trade-off (even ignoring energy usage).

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. 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. 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 the time cost of our approach. Obviously, this can vary significantly depending upon the particular combination of our schemes that are employed. As a first baseline, let us consider updating all agents with all data evidence, in a fully connected communication graph with parallel communication capabilities. Using our basic scheme (but not doing anything special by assuming no previous communication among the agents), 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 can be as low as $((a-1)+1)P$ or aP. Thus, each agent has its potential updated after a time as low as $M + (2a-1)P$, which is less than the MSBN. The advantage comes from our ability to better support MAS parallelism and avoid agents sitting idle. 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 $\frac{m}{2}P > M > P$): $M + (m + 4(a-1) + 2)P$ or $M + (m + 4a - 2)P$. Again, our approach will produce an interpretation faster, even though it does more computation, because the communication time plays a very small role due to parallelism. In fact, if additional data were continuously being received, communication time would become insignificant.

The above scenarios do not represent the intended usage of our approach, with limited, focused communication among agents, for approximate SI. Furthermore, if all data must be processed and used in interpreting every event, then large-scale SI will simply be impractical. Instead we would hope that events can be determined reliably enough based on only a reasonable fraction of the global data. When this is the case, it is likely that different subsets of the data would provide the most reliable support for each of the events. If we build our system so that each agent is responsible for only a subset of the global events (e.g., one agent responsible for determining whether E_1 is true or not, another agent whether E_2 is true or not, etc.), then we want different agents to base their answers on different subsets of the global data. This is exactly what our framework was designed to support.

The time and solution quality performance of such strategies will obviously depend on the specifics of each SI domain, and determining cost expressions for this type of strategy can be difficult. However, if we assume that all agents both receive and send data from the fraction f of other agents, the time for this strategy is approximately: $M+(m+4fa)P$. If f is on the order of 0.25, the time cost becomes approximately $M + (m + a)P$, while even if f is 0.5 (50% of the data is required) the time cost is $M+(m+2a)P$. Comparing these formulas to those above, we can see that this is much faster than would be possible with the MSBN (since M is much larger than P typically). So in appropriate domains, our framework could be used to implement much more efficient strategies than would be possible with the MSBN approach, while still providing acceptably high quality solutions.

5. ACKNOWLEDGMENTS

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

6. REFERENCES

[1] C. Huang and A. Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 11:1–158, 1994.

[2] Y. Xiang. A probabilistic framework for multi-agent distributed interpretation and optimization of communication. *Artificial Intelligence*, 9(3):171–220, 1996.

[3] Y. Xiang and V. Lesser. On the role of multiply sectioned bayesian networks to cooperative multiagent systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A*, 33(4):489–501, 2003.

[4] Y. Xiang, D. Poole, and M. Beddoes. Multiply sectioned bayesian networks and junction forests for large knowledge-based systems. *Computational Intelligence*, 87(1–2):295–342, 1993.