Controlling Information Exchange in Distributed Bayesian Networks*

Jiaying Shen and **Victor Lesser**
140 Governor's Drive
Department of Computer Science
University of Massachusetts
Amherst, MA 01003-4610, USA
{jyshen, lesser}@cs.umass.edu

Norman Carver
Computer Science Department
Southern Illinois University, mailcode 4511
Carbondale, IL 62901, USA
carver@cs.siu.edu

Abstract

We propose an algorithm that given a problem formulated as a Distributed Bayesian Network, finds a coordination strategy which minimizes the communication costs while achieving the desired confidence level of the global solution. We developed a system based on this algorithm which models the communication decision process for any given problem structure as a Markov Decision Process and use dynamic programming to produce the optimal communication strategy. To reduce the computational cost of the MDP approach, we further propose an algorithm based on the concept of Mutual Information to approximate the optimal solution. Experimental results for both systems are given to illustrate the effectiveness of the algorithms.

Introduction

In complex distributed applications, such as distributed interpretation, the amount of communication among agents required to guarantee global optimality or global consistency may be very significant. Thus, "satisficing" approaches have been developed that trade off guarantees of optimality for reduced communication. One approach is for agents to generate local solutions based on their own data and then transmit these high level solutions to other agents. Based on consistency and credibility of these local solutions, new local solutions may be generated or more detailed data sent until a sufficient level of consistency and credibility has been achieved among the agents. An important characterization of such distributed protocols is how much communication is required and the likelihood that the desired solution will be the same as what would be generated by an optimal centralized algorithm which used all available information.

Most approaches to managing communication trade off solution quality for reducing communication, but only from a statistical view. The behavior of the algorithms are often analyzed over an ensemble of problems to say that p percent of the time they will get the required solution quality q with an average amount of communication c (Carver & Lesser 2002).

*This material is based upon work supported by the National Science Foundation under Grant No. "NSF/CSS" (w/Carver@SIU); eff. 9/1/00-8/31/03: IIS-0004112

We would like to take this satisficing approach to the next step by exploring whether we can design a parameterized algorithm where we can predict, for a fixed amount of communication, the maximum level of confidence we expect in the final solution, or conversely, given a desired confidence level in the final solution how much communication the agents need. Finally, the algorithm should produce a communication strategy that will require only the minimum amount of communication to still achieve the desired solution quality.

We will study these issues in terms of Distributed Bayesian Networks (DBNs). Recent work includes algorithms (Xiang 1996) that produce the same final solution as is generated by a centralized problem solving system. However, this approach can potentially require significant communication.

In our research, we use a two-layer DBN to represent the underlying structure of the problem that needs to be solved as shown in Figure 1. For our problem, we make the following assumptions:

(1) There are two agents in the system.

(2) Every agent has access to the complete DBN.

(3) Evidence is distributed among the agents.

(4) Each agent knows what evidence the other agent has access to.

Bayesian Networks are a powerful tool to calculate conditional probabilities, and we have developed an algorithm that can reason about remote data using DBNs. Without exchanging any information at all, an agent can use our algorithm to compute the likelihood of a hypothesis H being the globally optimal solution based on its local data and direct the agent to ask for critical information from the remote agent in order to reach a higher level of confidence in the global solution.

We are proposing two approaches to generating communication strategies for any DBN problem structure based on this algorithm. With the MDP approach, given a DBN problem structure an agent will be able to dynamically construct a Markov Decision Process (MDP) and learn the optimal policy in terms of what data to ask for from the remote agent and in what order. On the other hand, the Mutual Information approach generates a myopic communication strategy which is not guaranteed to be optimal but requires less computational power and is more suitable for a larger and more

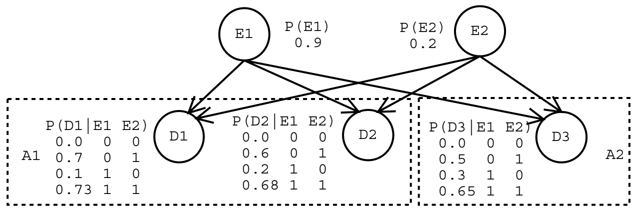

Figure 1: An example of a DBN problem structure. There are two events E_1 and E_2. Data D_1, D_2 and D_3 are distributed to two agents. A_1 has access to D_1 and D_2, while A_2 can only see D_3. The objective is for A_1 and A_2 to figure out what E_1 and E_2 are without too much communication.

complex DBN structure. With the smart conversation thus carried out by the agents, we expect the communication cost of the system to be significantly reduced.

Reasoning about Remote Data

In understanding our approach, let us first contrast it with the "satisficing" approach developed by Carver and Lesser for a distributed Bayesian Network to decide when and what to communicate. They call their strategy Consistent Local Solutions Strategy (CLSS) (Carver, Lesser, & Whitehair 1996). According to this strategy, agents first independently solve their local subproblems and then transmit their local solutions to all other agents. If these agents' local solutions are consistent with each other, they are merged without further verification of what the globally optimal solution is. If the local solutions are not consistent, lower level results/data are transferred to ensure that the local solutions chosen are consistent.

There are several key problems with CLSS. First, there is no way to ensure that the solution chosen is the globally optimal solution or that it has reached the desired confidence level. Second, there can be significant delays in problem solving when agents require substantial amounts of raw data from other agents. Hence, we propose the idea of transmitting data incrementally until sufficient confidence in the current best solution is reached. By "incrementally" we mean that not all of the local raw data is transferred at once. Instead, it is transmitted as needed until global consistency is achieved. This raises another question: in what order should the raw data be transmitted when it is necessary?

Instead of giving a one-size-fits-all strategy, we have designed an algorithm that can produce a strategy for any given DBN that requires as little communication cost as possible to achieve the desired confidence level of the final global solution.

We use a two-layer Bayesian Network to represent the problem structure (Figure 1). The top level nodes are the events that are the cause of the observed data, while the leaves are the raw data gathered, which are distributed to various agents. The objective of the system is to figure out the likelihood of events E_1 and E_2 without significant communication.

Definition 1 Agent A_i. *In Figure 1, there are two agents* $\{A_i | i = 1, 2\}$.

Definition 2 Event E_i. *The possible events in the environment which cause the observed data. For example, in Figure 1 there are two possible events* $\{E_i | i = 1, 2\}$.

Definition 3 Data D_i. *The data observed by agents. In Figure 1, we have 3 data* $\{D_i | 1 \le i \le 3\}$.

Definition 4 Evidence ϵ_i. *The data values possibly observed by one agent or later transferred from some remote agent. They are possible configurations of the data set. For example, in Figure 1 agent* A_1 *has four possible evidence configuration* $\{\epsilon_i | 1 \le i \le 4\}$ *when it does not have knowledge of any remote data, as shown in (1).*

$$\begin{array}{ccccc} & \epsilon_1 & \epsilon_2 & \epsilon_3 & \epsilon_4 \\ D_1 & 1 & 1 & 0 & 0 \\ D_2 & 1 & 0 & 1 & 0 \end{array} \tag{1}$$

Definition 5 Actual evidence ϵ_{A_i}. *Every agent can only know one data value configuration from the possible evidence set. In Figure 1,* A_1 *initially can only observe* $\epsilon_{A_1} \in \{\epsilon_i | 1 \le i \le 4\}$. *As communication goes on, an agent will gather more evidence from the remote agents, and the set* ϵ_{A_i} *will grow.*

Definition 6 Hypothesis H_i. *The possible hypotheses the agents might draw from the evidence. Normally they are possible configurations of the events. For example, in Figure 1 we have four possible hypotheses* $\{H_i | 1 \le i \le 4\}$, *as shown in (2).*

$$\begin{array}{ccccc} & H_1 & H_2 & H_3 & H_4 \\ E_1 & 1 & 1 & 0 & 0 \\ E_2 & 1 & 0 & 1 & 0 \end{array} \tag{2}$$

Definition 7 Likelihood L_{A_i}. *Based on the evidence observed, an agent can calculate the conditional probabilities of every hypothesis based on the evidence, i.e.,* $L_{A_i}(H) = P(H | \epsilon_{A_i})$.

Definition 8 MAPI (Maximum A Posteriori Interpretation). *Based on the current evidence set, there is a most likely interpretation of the events, i.e., a most likely hypothesis.* $MAPI(\epsilon) = argmax_H P(H | \epsilon)$.

Let us now assume that A_2 has not received any information from A_1. With the knowledge of its own data and the Bayesian Network structure, A_2 can still do some reasoning to decide what information it needs to determine the globally best hypothesis. We illustrate this with an example.

In our example, A_2 has access only to $D_3 = 0$ and the BN structure shown in Figure 1. Does A_1 need to send all the data? What information should A_2 request from A_1? Can A_2 save communication cost by requesting only the necessary data? Naturally, A_2 will put itself in the place of A_1 to reason about A_1's data. It calculates the probabilities of the four possible evidence of A_1 given $D_3 = 0$ as follows:

$$\begin{aligned} P(\epsilon_1 | D_3 = 0) &= 0.0679 \\ P(\epsilon_2 | D_3 = 0) &= 0.0987 \\ P(\epsilon_3 | D_3 = 0) &= 0.1632 \\ P(\epsilon_4 | D_3 = 0) &= 0.6701. \end{aligned} \tag{3}$$

	ϵ_1	ϵ_2	ϵ_3	ϵ_4
H_1	0.872	0.576	0.346	0.061
H_2	0.065	0.364	0.625	0.932
H_3	0.063	0.059	0.029	0.007
H_4	0	0	0	0

Table 1: What to request: $P(H_i|\epsilon_j \wedge D_3 = 0)$

A_2 can also calculate the probabilities of the four hypotheses assuming the evidence of A_1, i.e. $P(H_i|\epsilon_j \wedge D_3 = 0), 1 \leq i \leq 4, 1 \leq j \leq 4$, as in Table 1.

Definition 9 Compound Probability $CP(\epsilon_i|\epsilon_A)$ *is a pair* $(P(\epsilon_i|\epsilon_A), H)$, *where* $H = MAPI(\epsilon_A \wedge \epsilon_i)$. *It indicates that given the local evidence* ϵ_A, *with probability of* $P(\epsilon_i|\epsilon_A)$, H *is the best hypothesis, where* ϵ_i *are all the possible evidence of the remote agent.*

Definition 10 Confidence $C_A(H|\epsilon_A)$. *The likelihood of a hypothesis* H *being the MAPI of the whole evidence set given the current known evidence of agent* A. $C_A(H) = \sum_{i \in \{k|H=MAPI(\epsilon_k \wedge \epsilon_A)\}} P(\epsilon_i|\epsilon_A)$.

In our example, from (3) and Table 1, A_2 has the compound probabilities of the four possible remote evidence as follows:

$$\begin{aligned} CP(\epsilon_1|D_3 = 0) &= (0.0679, H_1) \\ CP(\epsilon_2|D_3 = 0) &= (0.0987, H_1) \\ CP(\epsilon_3|D_3 = 0) &= (0.1632, H_2) \\ CP(\epsilon_4|D_3 = 0) &= (0.6701, H_2). \end{aligned} \quad (4)$$

It is obvious that based on A_2's data, H_2 is the most probable hypothesis. However, from (4), we can see that for ϵ_1 and ϵ_2, H_1 is the best hypothesis, while for ϵ_3 and ϵ_4, H_2 is. The conclusion is thus: with probability of $P(\epsilon_1|D_3 = 0) + P(\epsilon_2|D_3 = 0) = 0.1666$, H_1 is the globally optimal hypothesis, and with probability of $P(\epsilon_3|D_3 = 0) + P(\epsilon_4|D_3 = 0) = 0.8333$, H_2 is the globally optimal hypothesis. So, we are able to collapse (4) into:

$$\begin{aligned} CP(\epsilon_1 \vee \epsilon_2|D_3 = 0) &= (0.1666, H_1) \\ CP(\epsilon_3 \vee \epsilon_4|D_3 = 0) &= (0.8333, H_2), \end{aligned} \quad (5)$$

From the compound probability, we get the confidence of $H_1 = 0.1666$ and $H_2 = 0.8333$.

If we collapse (1) into:

$$\begin{array}{ccc} & \epsilon_1 \vee \epsilon_2 & \epsilon_3 \vee \epsilon_4 \\ D_1 & 1 & 0 \end{array}, \quad (6)$$

it is easy to see that D_1 is what makes the difference of choosing H_1 or H_2 as the globally optimal hypothesis. Consequently, A_2 will be able to determine the globally optimal hypothesis if A_1 sends it the observed data D_1. Based on this knowledge, A_2 only needs to ask A_1 for D_1 instead of both D_1 and D_2. This reduces communication cost.

What is more, from (5) we can see that with probability of 0.8333, H_2 is the globally best hypothesis. As a result, if we only need to reach the confidence level of 80%, A_2 does not even need to request A_1 to send it data D_1. Only when the confidence requirement is above 83%, is additional data necessary. The compound probabilities enable the agent to see what data to request to achieve the desired confidence level and to communicate as little as needed.

Please note that our definition of confidence is unique in the sense that it not only takes into consideration the local evidence, but also the unknown remote evidence. Other work generally use MAPI as the measure of the confidence level. Definition 10 is not measuring the belief in a hypothesis as MAPI is, but how likely a hypothesis would be chosen as MAPI if all the evidence were available. We use this more complex form of confidence in order to take advantage of the fact that we can gain information about the remote data given our current knowledge, which reduces the uncertainty of the high level hypothesis.

Now we can summarize the algorithm as follows:

Algorithm 1
1. Calculate the probabilities of the possible evidence of the remote agent A_1 based on the local data, i.e. $P(\epsilon_i|\epsilon_{A_2})$.
2. Calculate the probabilities of the hypothesis assuming the remote evidence, $P(H_i|\epsilon_j \wedge \epsilon_{A_2})$.
3. Calculate and collapse the compound probabilities to get confidence of different hypotheses.
4. Group the evidence according to the most probable optimal hypothesis, and collapse the evidence table.
5. Request data according to the collapsed evidence table and the required confidence level of the final solution.

The advantage of Algorithm 1 is evident. With CLSS, A_1 will have to transmit both raw data D_1 and D_2 to get a globally consistent solution, while using Algorithm 1 we need to transmit only D_1 to ensure the globally optimal solution or no communication at all to reach the 80% confidence level.

One thing worth noting is that we assume that only A_2 is doing the reasoning in our algorithm. This is reasonable since there will often be one agent who is responsible for assembling the global solution. A more interesting case is when A_1 is simultaneously reasoning about what data it should provide to A_2. We will look into it in our future work.

An easy extension to this algorithm is for the remote agent to transmit its current MAPI (i.e. local solution) before sending any low level data. This knowledge will help the agent to get more information of the remote evidence, and thus influence the communication decision. For example, in Fig 1, if A_1 tells A_2 that its local MAPI is H_1, A_2 can reason about A_1's part ofthe DBN as if it were A_1 and do the inference of all the evidence of ϵ_1 to ϵ_4. It will thus get the following:

$$\begin{array}{ll} MAPI(\epsilon_1) = H_1, & MAPI(\epsilon_2) = H_1 \\ MAPI(\epsilon_3) = H_2, & MAPI(\epsilon_4) = H_2 \end{array} \quad (7)$$

Based on this calculation and the local solution of A_1, A_2 can easily eliminate the cases of A_1 observing ϵ_3 and ϵ_4, and make our later calculation much easier. This can be further extended to any data in the hierarchy.

Deriving Optimal Communication Strategies

Algorithm 1 has answered the question of what to request (communicate). Now, we have another equally important question to answer: if there is more than one piece of critical data, in what order and combination should the data be transmitted to minimize the communication cost. This is essentially a solution to step 5 of the algorithm.

To answer this question we frame the problem as an MDP and use dynamic programming to find an optimal communication strategy. Each state of the MDP includes the current known remote evidence, the current best solution and its compound probability, i.e.,

$$S : \epsilon_{k-r}, (C(H|\epsilon_l \wedge \epsilon_{k-r}), H) \tag{8}$$

where ϵ_{k-r} is the known remote evidence, ϵ_l is the local evidence, and H is the current best solution, i.e. $H = argmax C(H|\epsilon_l \wedge \epsilon_{k-r})$. For example, in Fig 2, we have the state $S1 = \{D_1 = 1\}, (1, H_1)$. In this state, agent A_2 knows that the actual value of the remote data D_1 is 1. Based on this information and its local evidence, the current best solution is H_1 with confidence of 1. The action set of the MDP is all the possible combinations of the critical data. The cost of each state-action pair is the amount of communication needed to take this action in this state. We assume that the cost of a request message is 1 no matter how many data are requested, and each data transmitted costs 1. The MDP starts at the state where no remote data is known and the best global solution is based on its own local information. It stops when the desired confidence level is reached.

As an example, we will construct an MDP (Fig 2) for the problem illustrated in Fig 1, which we have studied so far. Without knowing any remote data, A_2 may determine that the best global solution is H_2 with the confidence of 0.83. Going through Algorithm 1, it decides that the critical remote data set is $\{D_1\}$. As a result, the action set for A_2 is $\{(D_1)\}$. If it takes the action (D_1), it will get the globally optimal solution no matter what the reply is. Hence, it is a very simple MDP that we constructed, with only one start state and two final states. To obtain the confidence of 1, agent A_2 only needs to request data D_1.

This is a fairly simple example. To further illustrate more complex cases of our algorithm, let us now consider the problem illustrated below in (9) and the corresponding MDP shown in Fig 3. Please note that (9) is different from (4). In (9) the best hypothesis for ϵ_4 is H_3 instead of H_2. All the other assumptions are essentially the same as the previous example. The existence of H_3 makes our example sufficiently complex to better illustrate the algorithm.

$$\begin{aligned} CP(\epsilon_1|D_3 = 0) &= (0.0679, H_1) \\ CP(\epsilon_2|D_3 = 0) &= (0.0987, H_1) \\ CP(\epsilon_3|D_3 = 0) &= (0.1632, H_2) \\ CP(\epsilon_4|D_3 = 0) &= (0.6701, H_3). \end{aligned} \tag{9}$$

Without knowing any remote data, A_2 may determine that the best global solution is H_3 with the confidence of 0.67. Going through Algorithm 1, it decides that the critical remote data set is $\{D_1, D_2\}$. As a result, the action set for A_2 is $\{(D_1), (D_2), (D_1, D_2)\}$. If it takes the action (D_1, D_2), it will get the globally optimal solution no matter what the reply is. If it asks for D_1, with a probability of $0.0679 + 0.0987 = 0.1666$, $D_1 = 1$ and H_1 is the

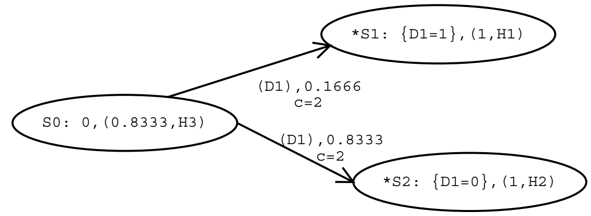

Figure 2: The MDP generated for the problem in Fig 1. Every state includes the information of the known evidence, and the CP value. Every transformation arrow denotes the action taken, the probability of getting to the next state and the cost of the action. S0 is the start state and the starred states are terminal states.

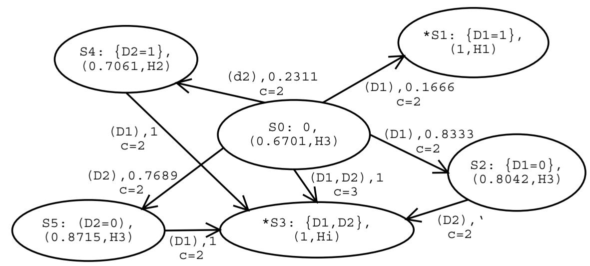

Figure 3: The MDP generated for the problem in (9). The final state S3 has been abbreviated to save space.

globally optimal solution with confidence of 1. If $D_1 = 0$, the remote data can be either ϵ_3 or ϵ_4. As a result, A_2 can decide only that H_3 is the best solution with confidence of $0.6701/(0.1632 + 0.6701) = 0.8042$. If it still wants to improve its confidence level, it will need to take further action, asking for D_2, after which it can draw the best conclusion with full confidence. The cost for path $S0 \rightarrow S3$ is 3 and for path $S0 \rightarrow S1$ is 2, while the cost for path $S0 \rightarrow S2 \rightarrow S3$ is 4. Applying Dynamic Programming to this MDP, to achieve confidence level of 80%, the best strategy is for A_2 to ask for only D_1, while to achieve confidence level of 100%, A_2 should ask for both D_1 and D_2 at once.

We have implemented a system to dynamically construct an MDP for any given problem structure based on this algorithm. The input of the system is the problem structure represented in the form of a DBN, and the output is the optimal communication strategy the agent should deploy. A BN toolkit is used to calculate all of the necessary conditional probabilities (steps 1–3 in Algorithm 1) and a decision tree is employed to collapse the truth table and find the data critical to the globally optimal solution according to Algorithm 1 (steps 3 and 4). The system then uses dynamic programming to produce the optimal communication action sequence for the MDP constructed for step 5.

A Myopic Approach

The obvious shortcoming of our MDP framework is its computational complexity. It does not scale well as the number of agents increases or as the complexity of the DBN structure grows. To solve this problem, a common method is to approximate the optimal solution.

We have noticed from the last two sections that different data are of different importance to reduce our uncertainty of the most likely hypothesis. In Fig 1, data pieces like D_1 are the most valuable as their values can reduce the hypothesis candidates. Though knowing the actual value of D_2 does not help to identify which hypothesis is the most likely one, it does help to raise the confidence level of the most likely hypothesis if it is identified. There might also be completely irrelevant data that can be ignored. Hence, it is important to find a suitable measure to identify the importance level of data.

Sensitivity matrices (Pearl 1988) are often used to measure the influence of a variable on a target variable. What we mean by "em influence" is how knowing the value of a variable can reduce the uncertainty of the target variable.

Definition 11 *Let ϵ be the current available evidence, T be the variable representing the target hypothesis, and let X be the test variable, i.e., an observable node whose impact on T is to be assessed. The* **sensitivity** *of $T = t$ to $X = x$ is often defined as*

$$S(t,x) = \frac{BEL(t|x)}{BEL(t)} = \frac{P(t|x \wedge \epsilon)}{P(t|\epsilon)}. \tag{10}$$

The sensitivity matrix $S(t,x)$ itself contains only the influence of every value of $X = x$ on every value of $T = t$ but does not provide a concise summary of the overall contribution of X to reducing the uncertainty in T. Mutual information (Pearl 1988) is one of the most commonly used measures for solving this problem and ranking information sources.

Definition 12 Mutual Information *$I(T,X)$ is used to measure the total uncertainty-reducing potential of X on the target variable T. It is defined as:*

$$I(T,X) = -\sum_x \sum_t BEL(t,x) \log \frac{BEL(t,x)}{BEL(t)BEL(x)}. \tag{11}$$

$I(T,X)$ is a nonnegative quantity and is equal to 0 if and only if T and X are mutually independent (Gallager 1968). The larger $I(T,X)$ is, the more important X is in reducing the uncertainty in T.

In our research, the main task is to find a communication strategy that will direct the low level data transfer between the agents in order to decide which high level hypothesis is the most likely one. The natural heuristic is to transfer the data with greater influence on reducing the uncertainty when deciding the best hypothesis. Mutual Information seems to be a good measure to complete this task.

The first thing we need to do is to define a suitable target variable for our measurement. We only care about the hypothesis H_i where $i = argmaxC(H_i|\epsilon)$, and ϵ is the currently available evidence. The importance of an unknown piece of data depends on how effective it will be in reducing the uncertainty of choosing H_i. As a result, we define the target variable in our system as follows:

Definition 13 Target Variable T *is the variable whose uncertainty the agents are trying to reduce.*

$$T = argmax_i C(H_i|\epsilon) \tag{12}$$

As we see that the value of T depends on the information ϵ available to the agent, it will change when the agent gains more information. As an example, let us again have a look at the problem illustrated in Fig 1. In this example, there are four mutually exclusive hypotheses $\{H_i|1 \leq i \leq 4\}$ as listed in (2). The target variable T for agent A_2 should be: $T = i$, if H_i is the most likely hypothesis given the current information known to A_2.

If A_2 has access to $D_3 = 0$ only, and is trying to decide what data to request from A_1, let us first see what T should be before and after we assume the value of D_1. Let us write the T value before assuming the value of D_1 as T_b and that after as T_a. Before assuming anything about D_1, the only evidence A_2 has is $D_3 = 0$, hence $\epsilon = \{D_3 = 0\}$, and we have

$$T_b = argmax_i C(H_i|D_3 = 0), \tag{13}$$

On the other hand, if A_2 assumes that $D_1 = d_1$, the evidence will become $\epsilon = \{D_1 = d_1 \wedge D_3 = 0\}$. As a result, it will recalculate the value of T as

$$T_a = argmax_i C(H_i|D_1 = d_1 \wedge D_3 = 0). \tag{14}$$

Now we can calculate the mutual information values for D_1 and D_2 in regard to T as in (15), where $i = 1, 2$:

$$I(T,D_i) = -\sum_{d_i} \sum_t P(T_a = t, D_i = d_i|D_3 = 0) \times$$
$$\log \frac{P(T_a = t, D_i = d_i|D_3 = 0)}{P(T_b = t|D_3 = 0)P(D_i = d_i|D_3 = 0)} \tag{15}$$

We have previously calculated Compound Probabilities $CP(D_1, D_2|D_3 = 0)$ as in (4), from which we can get mutual information for D_1 and D_2 as follows:

$$I(T,D_1) = 0.20, \qquad I(T,D_2) = 0.006 \tag{16}$$

Clearly, $I(T,D_1) > I(T,D_2)$, which coincides with the conclusion in the last section well that D_1 is more important than D_2. Hence, agent A_2 will request data D_1 from A_1. After receiving $D_1 = 1$, A_2 can again calculate $I(T,D_2)$ with the new $T_b = argmax_i P(H_i|D_1 = 1 \wedge D_3 = 0)$ and $T_a = argmax_i P(H_i|D_1 = 1 \wedge D_2 = d_2 \wedge D_3 = 0)$. Not surprisingly, $I(T,D_2) = 0$, which means that T and D_2 are independent of each other and thus D_2 will not help in reducing the uncertainty of the global solution.

Now that we have shown how Mutual Information can help us decide which remote data to request, we propose the following algorithm to calculate a **Myopic Communication Strategy** for any given problem structure:

Algorithm 2 *1. For every unknown remote data d_i, compute $I(T,d_i)$.*

2. *Choose d_i with the largest $I(T,d_i)$, and ask the remote agent to transmit it.*

3. *Propagate the newly arrived evidence d_i. If the likelihood requirement is met, stop. If not, go to step 1.*

We call this algorithm *myopic* because it only looks one step ahead when the agent is deciding which data to request in contrast with the MDP approach. It is not considering whether it is the optimal decision in the long run.

It may also be noticed that in the algorithm we formulated above, the possible actions are only requesting single pieces of data. It can be easily extended to multi-data request if we define the test data as any combination of the unknown remote data.

The uniqueness of our algorithm lies in the definition of the target variable. As we have discussed before, the utilization of our new definition of "confidence" enables us to inference about the high level hypothesis not only based on the known evidence but also on the unknown remote evidence.

Experimental Results

We have implemented a communication strategy system that generates both the optimal policy with an MDP and the myopic policy with Mutual Information. We have done experiments on 50 different DBN structures. Every DBN structure includes 5 top level events and 10 lower level data pieces. The data pieces are evenly distributed to 2 different agents with no overlapping. For each structure, the system has the input of a required confidence level of the final solution. After calculation it will output both strategies and the minimum percentage of the remote data that has to be transferred in order to obtain the required confidence level. We average the amount of data communicated over the 50 structures and get the graph as shown in Fig 4. The graph shows that when the required confidence is as low as 60%, only about 10% of the data need to be transferred, while as the confidence requirement increases to 1, the communication cost jumps up to over 70%. Nevertheless, both strategies only need a little over 70% of the data to be communicated to be able to achieve the confidence level of 1, which is very encouraging. Furthermore, unsurprisingly the policy generated with MDP uniformly performs better than that with Mutual Information, but not by a substantial amount. The size of the MDPs generated for the DBN structures is on the order of 300. We have also done experiments on DBN of different sizes. The size of the MDPs grows exponentially with the number of data pieces in the DBN while the number of steps required by the Mutual Information approach grows linearly. This demonstrates that the Mutual Information approach is a good approximation of the optimal approach while it has a much better scalability.

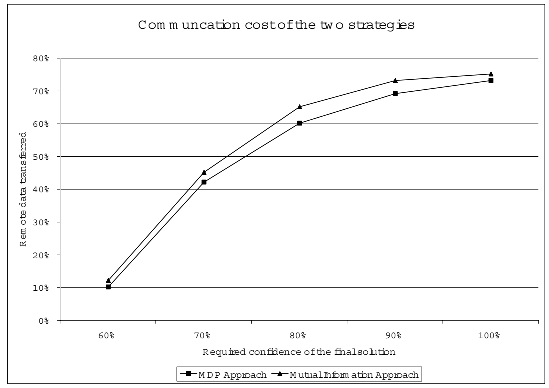

Figure 4: Experimental Results of the Communication Strategy System

Future Work

We plan to collect more experimental results and compare the results with the data collected from (Carver & Lesser 2002) without communication planning. Some formalization based on the statistics result will also be done to predict the amount of information that needs to be exchanged to reach a certain level of confidence. Furthermore, we will try to apply some approximation techniques to reduce the computational complexity inherent to Bayesian Networks. Dynamic programming will guarantee the optimality of the solution, but it is also time consuming and computationally expensive. We are considering applying some reinforcement learning techniques such as Q-learning to approximate the optimal policy. It would also be interesting to experiment on two-way communication strategies, and to investigate how the agents will interact with each other if both of them are active.

Scaling up the current algorithms is another area we are planning to work on. One of the directions is to extend our algorithms to more than two agents. One of the easiest way is to view all the other agents as the remote agent in the current algorithms. However there may be more efficient and interesting solution.

The reasoning about what data to request gives us some insight into the relationship between the confidence level in the hypothesis and the communication needed. In Carver (Carver & Lesser 2002), they presented some experimental results on this relation in the context of different near-monotonicity levels. This work may help us explain those results. We are currently looking into the relation between the near monotonicity measures used by them and the method used here, in hopes of finding such explanations.

References

Carver, N., and Lesser, V. 2002. Domain monotonicity and the performance of local solutions strategies for CDPS-based distributed sensor interpretation and distributed diagnosis. *to appear in International Journal of Autonomous Agents and Multi-Agent Systems*.

Carver, N.; Lesser, V.; and Whitehair, R. 1996. Nearly monotonic problems: A key to effective FA/C distributed sensor interpretation. In *Proceedings of AAAI-96*, 88–95.

Gallager, R. 1968. *Information theory and reliable communications*. John Wiley & Sons.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc.

Xiang, Y. 1996. A probabilistic framework for multi-agent distributed interpretation and optimization of communication. *Artificial Intelligence* 87(1–2):295–342.