Abstraction and Aggregation in Belief Networks

Ole J. Mengshoel and **David C. Wilkins**
Department of Computer Science and Beckman Institute
University of Illinois, Urbana-Champaign
Urbana, IL 61801
{mengshoe, wilkins}@cs.uiuc.edu

Abstract

Abstraction and aggregation are useful for increasing speed of inference in and easing knowledge acquisition of belief networks. This paper presents previous research on belief network abstraction and aggregation, discusses its limitations, and outlines directions for future research.

Introduction

Abstraction and aggregation have been used in several areas in artificial intelligence, including in planning, model-based reasoning, and reasoning under uncertainty. For reasoning under uncertainty, the framework of decision theory and in particular the notion of influence diagram (or decision diagram) has proven fruitful. An influence diagram is essentially a graph, where nodes are chance nodes, decision (or action) nodes, or utility (or value) nodes. This paper focuses on abstraction and aggregation in belief (Bayesian, causal, probabilistic) networks. Belief networks are influence diagrams restricted to contain only chance nodes (Pearl 1988) (Jensen 1996).

For the purpose of this paper, we will distinguish between abstraction and aggregation, which both have been described in the belief network (BN) literature (Chang & Fung 1991) (Wellman & Liu 1994) (Liu & Wellman 1996). *Abstraction* is essentially to replace several node states with one node state. Abstraction is also known as state-space abstraction (Wellman & Liu 1994) (Liu & Wellman 1996), coarsening (Chang & Fung 1991), or behavioral abstraction (Genesereth 1984). *Aggregation* is essentially to replace several nodes with one node. Aggregation is also known as structural abstraction (Wellman & Liu 1994) (Genesereth 1984) or hierarchical abstraction (Srinivas 1994). The inverse operations of abstraction and aggregation, refinement and decomposition, are also of interest; these will be discussed below as will other issues related to abstraction and aggregation.

The rest of this paper is organized as follows. First, we present previous research on abstraction and refinement; this research is then evaluated. Second, we focus on previous research concerning aggregation and decomposition. Third, we consider directions for future research in the areas of abstraction and aggregation.

Abstraction and Refinement

Previous Work

Fung and Chang introduced the two operations of refine and coarsen for BNs (Chang & Fung 1991). Coarsen eliminates states for a node (it is an abstraction operation), while refine introduces new states for a node (it is a refinement operation). Both operations take as input a target node and a desired refinement or coarsening, and they output a revised conditional probability distribution for the target node and for the target node's children. If V is a node (random variable), then Ω_V is its state space. In particular, we consider a node X and its parent P. The following are constraints that apply to the new conditional probability distribution when $x \in \Omega_X$ is refined to $x' \in \Omega_{X'}$. The first constraint applies to a node X and its parent P, where $p \in \Omega_P$:

$$\Pr(x \mid p) = \sum_{x' \in R(x)} \Pr(x' \mid p) \qquad (1)$$

Here, R is a refinement function that maps a single value $x \in \Omega_X$ into multiple values $x' \in \Omega$.

The second constraint applies to the node X, its child C, and the child's other parent Q. Here, $q \in \Omega_Q$ and $c \in \Omega_C$:

$$\begin{aligned} \Pr(c \quad \mid \quad & x, q) \Pr(x \mid p) \\ = & \sum_{x' \in R(x)} \Pr(c \mid x', q) \Pr(x' \mid p) \end{aligned} \qquad (2)$$

Based on these two constraints, external and internal refinement operations are defined. Similar constraints define external and internal coarsening as well. External operations change the BN topology by using Shachter's operations (Shachter 1988). Internal operations, on the other hand, maintain the BN topology. Fung and Chang define an internal abstraction operation, we will call it Fung and Chang (FC) abstraction.

Wellman and Liu investigate abstraction and refinement for anytime inference using BNs (Wellman & Liu 1994) (Liu & Wellman 1996). Their anytime algorithm is as follows. First, an initial abstract BN (ABN) is created from a ground BN (GBN). For all abstracted nodes, the ABN contains one superstate. Second, the ABN is used to compute the distribution of interest, using some BN inference algorithm. Third, if all states for all nodes have been refined, return; else refine a selected superstate. Fourth, go to the second step. The key point to note in this algorithm is that it can be interrupted after any iteration and provide an approximate result, and the approximation improves over time.

Wellman and Liu in their algorithm rely on abstraction of a node X, and in particular how a parent P and a child C are affected. Updating of $\Pr(X \mid P)$ is similar to Fung and Chang, while updating of $\Pr(C \mid X)$ is different. Let the states $x_i, ..., x_j \in \Omega_X$ be abstracted to $x'_{ij} \in \Omega_{X'}$. Wellman and Liu's abstraction operation (WL abstraction) approximates the conditional probability distribution and is defined as:

$$\Pr(c \mid x'_{ij}) = \frac{\sum_{l=i}^{j} \Pr(c \mid x_l)}{j-i+1},$$

where $|C(x')| = j-i+1$. In other words, the $j-i+1$ states $x_i, ..., x_j$ in node X are abstracted into one state x'_{ij} in node X'. The approach here is to approximate by averaging over all the conditionals $\Pr(c \mid x)$ of X.

Nicholson and Jitnah have studied abstraction and refinement empirically (Nicholson & Jitnah 1996). They use the stochastic simulation inference algorithm of likelihood weighting. The performance of likelihood weighting without and with abstraction is compared. The two abstraction operations considered are WL abstraction and CF abstraction. The BN used for comparison is the ALARM network. The following parameters were varied: (i) skewness, (ii) position and number of evidence and query nodes, and (iii) level of abstraction. Nicholson and Jitnah reports that CF abstraction generally gives more accurate results than WL abstraction; on the other hand CF abstraction is computationally more expensive than WL abstraction. This corresponds to results obtained by Liu and Wellman (Liu & Wellman 1996).

Evaluation of Previous Work

Abstraction and aggregation are essential for analysis of large organizational structures operating in complex, dynamic environments. Such organizational structures comprise many persons, objects, and entities that interact and cooperate. An example of a such organizational structure is a military unit. We consider abstraction and aggregation in the context of battlefield reasoning, and in particular intelligence analysis. Military intelligence analysts need to integrate reports from a wide spectrum of sensors and intelligence sources in order to appreciate what is going on on the battlefield. A typical report format is shown in Figure 2. As suggested in this figure, reports and sensory data are uncertain and can refer to entities of interest at different levels of abstraction. For example, a scout report might be about 'tanks' while some other sensor can only say 'vehicles', although in reality the same entities are referred to. This type of abstraction is well covered by previous research as discussed above (Chang & Fung 1991) (Liu & Wellman 1996).

An area where abstraction is more prominent and less explored in previous research is that of spatial abstraction. In intelligence analysis this concerns the location of enemy military units. There is substantial uncertainty associated with where a unit is at a particular time, and the state space of a node representing military unit position is very large.

Figure 3 shows a simplified terrain map illustrating the essence of the situation. The enemy has four units (Unit1, Unit2, Unit3, and Unit4) in positions towards the north as indicated on the terrain map. Friendly forces are in the south. The enemy forces are attacking, while the friendly forces are defending. From an intelligence point of view, a key element is location of enemy forces as the situation evolves. Consider a map covering an area of 100 km × 100 km. In a chance node L representing location of one unit, this would give a highest level of abstraction covering 100 km × 100 km using, say, grid squares of 10 km × 10 km. The ground level, on the other hand, could consist of 10 m ×10 m grid squares. With a uniform factor of 10 along both the north-south direction and the east-west direction, this yields an abstraction hierarchy for L consisting of 5 levels.

Uniformely subdividing the state space of a location node, as suggested above, is one of many possibilities. In general, there are issues related to the dimensions along which to abstract, the shape of grids, and the abstraction factor. In addition comes the fact that intelligence analysts are more interested in number of units in a set of locations rather than which units are where.

Another limitation of previous research is that the ABN is constructed from the GBN using some abstraction operation, be it CF abstraction or WL abstraction. In very large (and possibly infinite) state spaces such as the one mentioned above, it would be preferable not having to construct the ABN from the GBN. Rather, one should have a form of knowledge-based refinement that allows refinement without having performed abstraction first.

Aggregation and Decomposition

Previous Work

Previous research on aggregation has been less extensive than that on abstraction, even though in some sense the issues involved are more comprehensive: Aggregation makes 'larger' changes to the state space.

Time	Sensor	Location	Size	Equipment	Unit	Activity
200917	SIGINT	NK285105	?	artillery	?	traffic
200943	JSTARS	NK4300 to NK3000	200+	MTI	?	moving
201203	REMBASS strings	R2-R5	200+	vehicles	?	moving

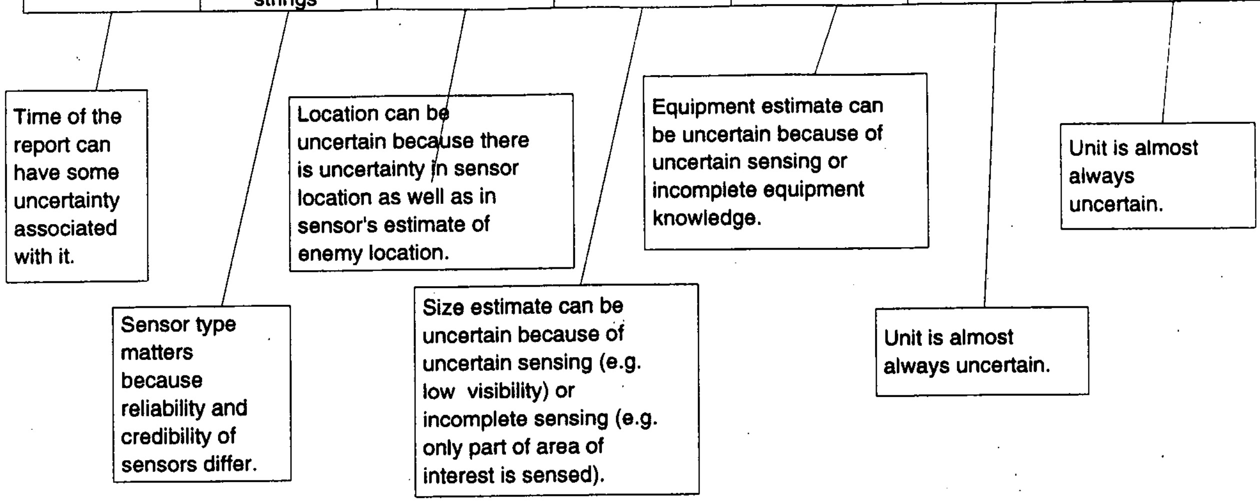

Figure 1: Format and features of reports for intelligence analysis.

Srinivas investigates the relationship between model-based reasoning and BNs (Srinivas 1994). Specifically, he constructed a translation algorithm that creates a BN from a hierarchical functional schematic, and describes how clustering (Lauritzen & Spiegelhalter 1988) (Jensen, Olesen, & Andersen 1990) can be modified to exploit the hierarchy. This work has been extended to include computation of repair plans (Srinivas & Horvitz 1995). The basic functional schematic unit is a component C. A component has input, output, and a mode; the mode represents operational status of the unit (e.g. correct, stuck-at-1, or stuck-at-0 for a transistor). Unless it is atomic, a component is recursively decomposed into subcomponents $C_1, ..., C_n$. The translation scheme takes a functional schematic and creates a BN consisting of these nodes: I_i represents input i, O represents the output, M_i represents mode i, and X_i represents internal variable i. At the level of subcomponents $C_1, ..., C_n$, we have a BN representing the joint distribution $\Pr(I_1, ..., I_m, O, M_1, ..., M_n, X_1, ..., X_k)$ At a higher level, C is considered an aggregate and the corresponding BN represents $\Pr(I_1, ..., I_m, O, M)$. Srinivas describes how the higher level distribution is computed from the lower level distribution using Shachter's topological transformation operations for marginalization (Shachter 1988).

By adding a dummy node D as a child to the nodes representing an aggregate component C, and a dummy node D' as a child to the input, mode, and output nodes representing the decomposition of C, an interface between the aggregate and the decomposed BN is established by adding an undirected arc between the join trees containing D and D' respectively. This means that computations from other parts of the BN affect the belief in C' only through this arc, and vice versa, making inference more localized and hence more efficient.

Wellman and Liu present some initial remarks on aggregation (Wellman & Liu 1994); in particular they note that aggregation, like abstraction, can be used for anytime inference. Anytime inference based on aggregation would start with a model consisting of highly aggregate BN nodes, and as time permits the BN is decomposed.

Evaluation of Previous Work

Aggregation is important in the intelligence analysis application. For example, a number of tanks make up a tank platoon. At a higher, more aggregate level, a brigade consists of three battalions plus a brigade staff. An interesting constraint in this domain is the hierarchical structure imposed by the chain of command. This restriction on an arbitrary BN could be exploited for both BN construction and inference, assuming that there are BN arcs that reflect the chain of command. On the surface, this is similar to Srinivas' hierarchical functional schematic. However, the military hierarchical structure is not isomorphic to the functional schematic hierarchical structure, and thus that work can not be immediately adopted. Another difficulty is

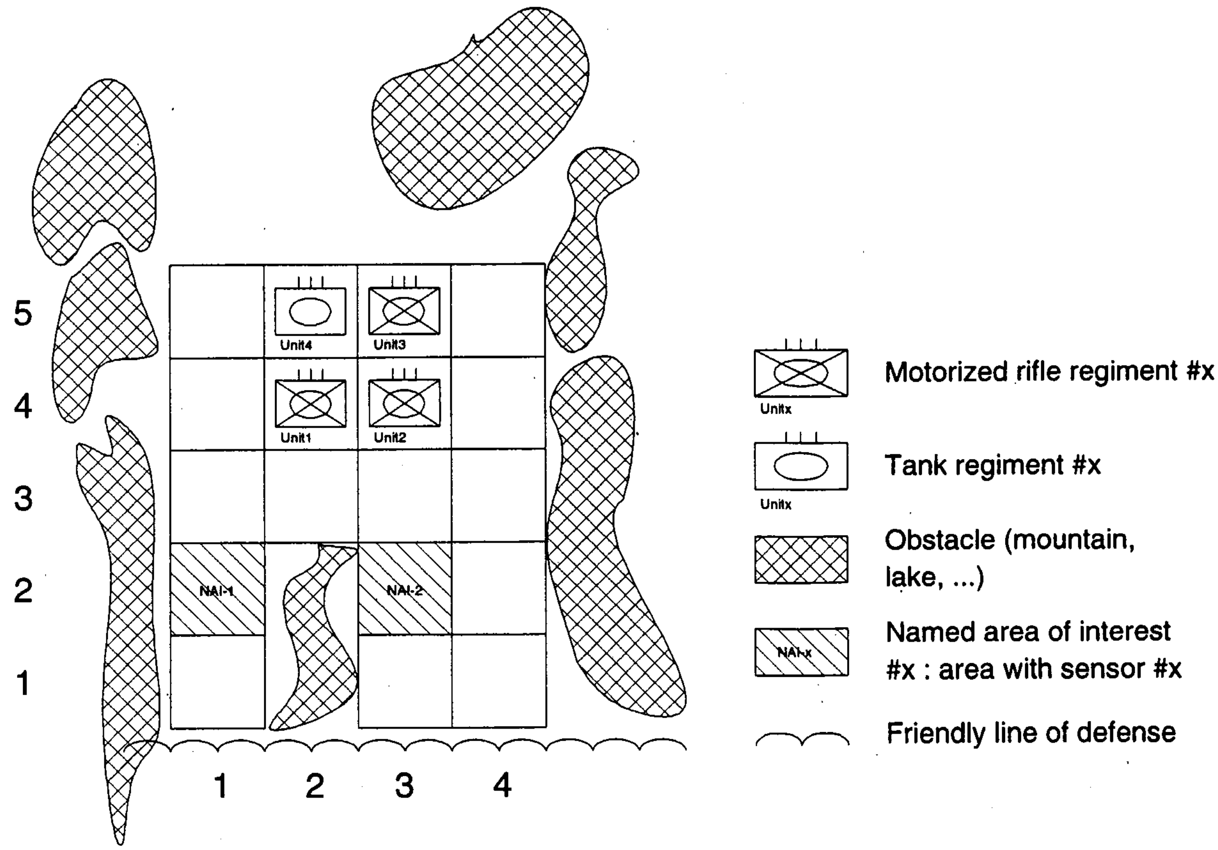

Figure 2: Simplified example terrain map.

the structure of observations for military analysis as opposed to for functional schematics. For the functional schematic, evidence is for a large part restricted to input and output nodes. For military analysis, evidence can concern almost all BN nodes. This means that using Srinivas' scheme, one would have to decompose a much larger part of a military analysis BN than a functional schematic BN, so there seems to be certain limitations in the Srinivas scheme when it comes to this type of evidence.

Several of the remarks made in the section on abstraction and refinement concerning the size of the space and the desirability of a top-down approach apply to aggregation as well. A brigade, for instance, consists of several thousand men and many hundred vehicles, and thus the most decomposed, atomic level will be unruly both with regard to knowledge acquisition and on-line abstraction.

Directions for Future Research

The summaries and evaluations of previous work has hinted at some directions for future research. This section makes additional recommendations.

Abstraction and Aggregation

Very little information on the relationship between abstraction and aggregation is found in the literature. This remark is among the exceptions (Wellman & Liu 1994):

> [Aggregation and abstraction] are complementary approaches to probabilistic-network approximation. They are also related, as abstracting the state space of a node to a single superstate is tantamount to ignoring the node.

An essential distinction between abstraction and aggregation is that the former assumes mutual exclusion, while the latter does not. This follows from the fact that abstraction and refinement operate on essentially the 'same' node, while aggregation and decomposition do not. This suggests that the two operations are orthogonal, supporting the quote above. However, when considering abstraction and aggregation in the military analysis domain, a high level of aggregation seems to suggest a high level of abstraction. For example, for a highly aggregated unit such as a brigade, highly abstracted locations appear to be most adequate. More research is needed to explore the relationship between abstraction and aggregation.

Abstraction and Aggregation in Dynamic Environments

Belief networks for dynamic environments make up an important class of BNs (Forbes *et al.* 1995) (Huang *et al.* 1994) (Nicholson 1992) (Provan 1993) (Dagum, Galper, & Horvitz 1992). Dynamic environments have at least two ramifications for belief network inference. First, the speed of inference is even more important than in the static case, since the environment is changing while inference takes place. Second, some form of propagation of the current environment state to the future needs to take place, at least in environments that are inaccessible, i.e. in cases where all relevant aspects of the environment are not available to the sensor nodes at all times. And most realistic environments, such as the ones inhabited by the large organizational structures mentioned earlier, are inaccessible. The issue of inference speed is addressed by the anytime algorithm research of Wellman and Liu. The issue of the role of abstraction and aggregation in state propagation has, to our knowledge, received little attention among BN researchers, and investigating this is an interesting research direction.

Abstraction and Aggregation in Dynamic BNs

Related to BNs for dynamic environments are dynamic BNs. Dynamic BNs are often used in dynamic environments, but can be applicable also in other cases (Charniak & Goldman 1993). In dynamic BNs, the BN topology typically changes over time (Provan & Clarke 1993) (Charniak & Goldman 1993). Some of these changes may be considered to be abstraction or aggregation, and by identifying them as such one would establish a firmer foundation for them. This would also facilitate comparison to other dynamic BN operations. Informally, an advantage of aggregation and abstraction versus arbitrary dynamic BN operations is that some 'residue' of the GBN is always maintained in the aggregated or abstracted BN, at least in an approximated form.

Features of refinement and decomposition could be exploited for dynamic BNs created by knowledge-based model construction. This has the potential of making the process of constructing a BN for inference top-down (using refinement and decomposition) rather than first bottom-up, then top-down (also using abstraction and aggregation). Abstraction and aggregation in dynamic BNs is a promising research direction.

Acknowledgements: This work was supported in part by ONR Grant N00014-95-1-0749 and ARL Grant DAAL01-96-2-0003.

References

Chang, K., and Fung, R. 1991. Refinement and coarsening of Bayesian networks. In *Uncertainty in Artificial Intelligence 6*, 435–445.

Charniak, E., and Goldman, R. P. 1993. A Bayesian model of plan recognition. *Artificial Intelligence* 64(1):53–79.

Dagum, P.; Galper, A.; and Horvitz, E. 1992. Dynamic network models for forecasting. In *Proceedings of the Eigth Annual Conference on Uncertainty in Artificial Intelligence (UAI-92)*, 41–48.

Forbes, J.; Huang, T.; Kanazawa, K.; and Russell, S. 1995. The BATmobile: Towards a Bayesian automated taxi. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, 1878–1885.

Genesereth, M. 1984. The use of design descriptions in automated diagnosis. *Artificial Intelligence* 24:411–436.

Huang, T.; Koller, D.; Malik, J.; Ogasawara, G.; Rao, B.; Russell, S.; and Weber, J. 1994. Automatic symbolic traffic scene analysis using belief networks. In *Proc. of AAAI-96*, 966–972.

Jensen, F. V.; Olesen, K. G.; and Andersen, S. K. 1990. An algebra of Bayesian belief universes for knowledge-based systems. *Networks* 20(5):637–659.

Jensen, F. V. 1996. *An Introduction to Bayesian Networks*. New York, New York: Springer Verlag.

Lauritzen, S., and Spiegelhalter, D. J. 1988. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society series B* 50(2):157–224.

Liu, C., and Wellman, M. P. 1996. On state-space abstraction for anytime evaluation of Bayesian networks. *SIGART Bulletin* 50–57.

Nicholson, A. E., and Jitnah, N. 1996. Belief network algorithms: A study of performance using domain characterisation. Technical Report 96/249, Department of Computer Science, Monash University.

Nicholson, A. E. 1992. *Monitoring Discrete Environments Using Dynamic Belief Networks*. Ph.D. Dissertation, Oxford University.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, California: Morgan Kaufmann.

Provan, G., and Clarke, J. 1993. Dynamic network construction and updating techniques for the diagnosis of acute abdominal pain. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(3):299–307. .

Provan, G. M. 1993. Tradeoffs in knowledge-based construction of probabilistic models. *IEEE Trans. on Systems, Man, and Cybernetics* 24(11):1580–1592.

Shachter, R. 1988. Probabilistic inference and influence diagrams. *Operations Research* 36:589–604. .

Srinivas, S., and Horvitz, E. 1995. Exploiting system hierarchy to compute repair plans in probabilistic model-based diagnosis. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*, 523–531.

Srinivas, S. 1994. A probabilistic approach to hierarchical model-based diagnosis. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, 538–545.

Wellman, M. P., and Liu, C.-L. 1994. State-space abstraction for anytime evaluation of probabilistic networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, 567–574.