The Use of Bayesian Networks in Critical Applications

R.D. Hawkins; University of York; York, UK

Prof. J.A. McDermid; University of York, UK

Keywords: Bayesian networks, safety

Abstract

Bayesian Belief Networks (BBNs) provide a way to model problems involving uncertainty. BBNs are directed acyclic graphs whose nodes are uncertain variables, and whose edges represent probabilistic dependencies between these variables. Each node has a conditional probability function associated with it, that models the relationship between the node and its parents. BBNs are used quite widely for diagnostics, particularly medical diagnosis, where they can be used to infer the most likely illness given data about a patient and their symptoms. BBNs may also be used to solve problems in other types of system where the relationship between variables is uncertain.

Many such applications will often require the use of BBNs in highly-dependable roles, where the information they provide could be critical to the safety of the system. In this paper we explore the issues associated with the use of BBNs in such critical roles, and suggest approaches to address these issues.

Introduction

The use of BBNs may be particularly useful in real-time control systems which often rely on information about the environment which is only partially observable, or may be noisy or incomplete. BBNs can be used to provide a way of inferring more accurate information about the state of such a system and its environment. Many real-time systems will often be safety critical in nature, meaning that failures of the systems could contribute to a hazard. In such situations the information provided by the BBN could be critical to the safety of the system. It is important therefore to be able to ensure that the system responds in a safe manner to all situations. In this paper we firstly describe a typical safety related application of BBNs. The challenges posed by using BBNs in safety critical applications are then discussed. The paper then goes on to report some initial work which suggests some potential approaches to addressing these issues. Finally, the work required to develop the approaches further is outlined, along with potential difficulties that will need to be resolved.

An Application of a BBN to a Real-Time Safety Critical System

In this section a simple example of a real-time BBN system is described which highlights many of the characteristics expected of such networks. This example will be used to highlight the safety issues associated with such systems. The example that shall be used is a highly simplified fire detection system, which uses information from a number of sensors to infer the type of fire. The basic structure of the BBN is shown in figure 1.

The fire type indicates the probability, or belief that the fire is in a particular state. For simplicity, there are just two fire states considered, fire, or no fire. The other four nodes in the network (figure 1) are different types of sensors which are used in detecting the current fire state. The arcs between the fire type node and the sensor nodes indicates that the fire type has an influence on the observations made by the sensors. It should be noted that none of the sensors are completely reliable, they all have a less than 100% probability of detecting a fire. This probability is captured in a Conditional Probability Table (CPT) for the relationship. Figures 2 and 3 show the network after compilation on the Netica tool (ref.1). Figure 2 shows the probabilities on the nodes when a fire is present (probability of fire state true is 100%). This illustrates the reliability of the various sensors. It should be noted that the probabilities on the sensor nodes are for illustrative purposes only. In reality these initial probabilities for the sensors would need to be determined based on testing and in-service experience. It should also be noted that if there were more fire types, rather than just fire and no fire, then the probabilities would vary depending on the current belief state for the fire type node. The more serious the fire, the more likely the sensors would be to detect it, so the probability of the sensor reading true would increase.

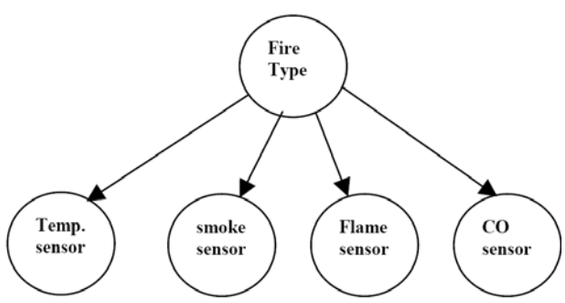

Figure1 – BBN for a Fire Detection System

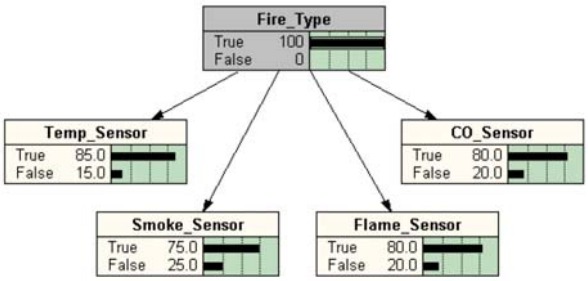

Figure 2 – Probability of Nodes when Fire Present

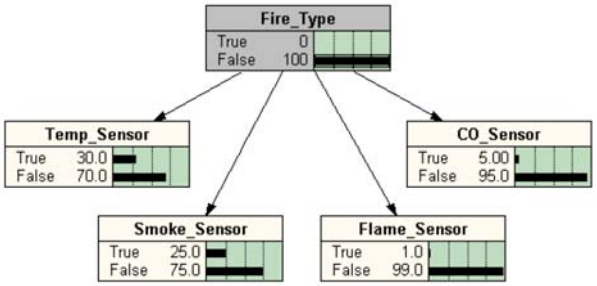

Figure 3 – Probability of Nodes when Fire not Present

Figure 3 shows the probabilities on the nodes when a fire is not present (probability of fire state false is 100%).The probabilities on the sensor nodes indicate the probabilities of receiving 'false positive' readings. Some sensors are more susceptible to this than others. Once again the figures used are purely for example purposes. During normal operation, it would not be known for certain whether there was a fire or not (indeed that is the whole motivation for the use of a BBN). Figure 4 shows an initial configuration for the network before any evidence is added. Without any evidence to the contrary, it is assumed that there is no fire, however since there is some uncertainty in this assumption a probability of 10% is assigned to fire. Once observations from the sensors are added, the probability of the belief state changes. Figure 5 shows the network when all sensors report false. If a sensor detects a fire then the belief that there is a fire increases. Figure 6 shows how the probability of fire increases when the flame sensor detects a fire. The effect on the belief state of the fire is dependant on the accuracy of the response of the sensor. The most responsive sensors increase the belief probability of fire the most. As more sensors also detect a fire the belief increases further. Figures 7 and 8 show the effect of more sensor results on the belief state.

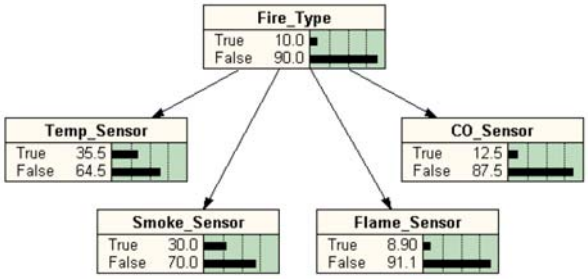

Figure 4 – Initial Configuration for BBN

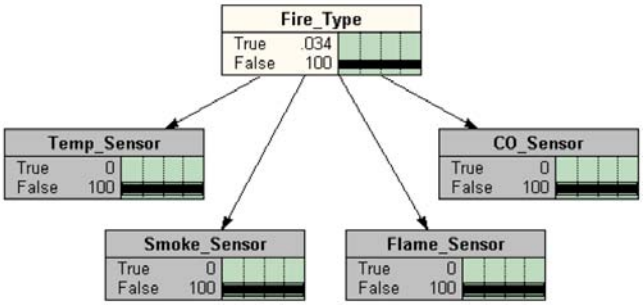

Figure 5 – BBN with all Sensors Reporting False

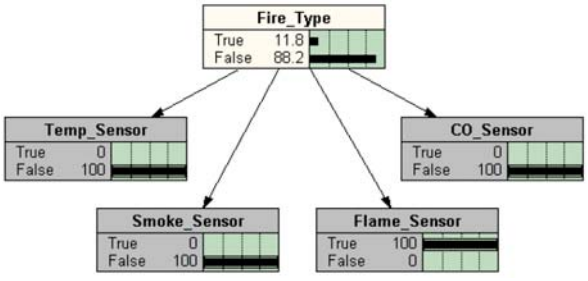

Figure 6 – BBN with Sensor Detecting Fire

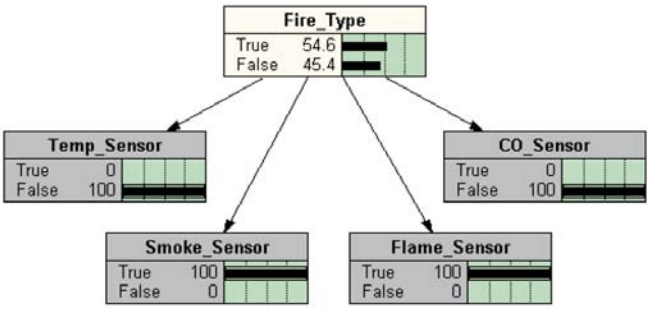

Figure 7 – More Sensors Detect Fire

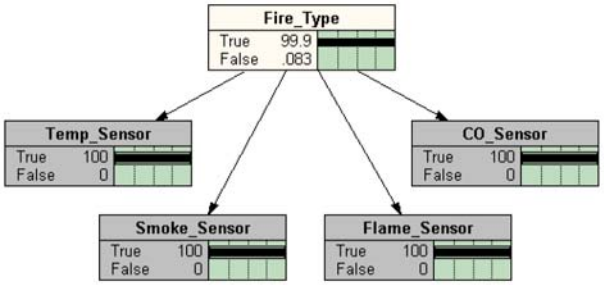

Figure 8 – All Sensors Detect Fire

Developing the Sensor Model

In the example above, it was assumed that the sensors were working correctly. The reliability of sensors may not be constant, over time sensors may degrade or fail. This would mean that the CPT associated with the sensor would no longer be valid. This would result in an incorrect belief state for the network. If the sensor has degraded or failed then the CPT must be altered to reflect this, such that the belief state takes into account the actual performance of the sensor. As shown in reference 2, the network can be used to dynamically estimate the reliability and accuracy of its sensors. By including a sensor status node for a sensor, it is possible to specify whether the sensor is performing well, at some degraded level, or has failed completely. The CPT for the sensor reflects the current sensor status. As variation between the predicted and observed values increases, the state of the sensor status node and the corresponding CPT will change to reflect this. In this way the network should maintain an accurate belief state even if sensors begin to degrade.

This example also uses a simple threshold system. That is to say that once a detected value, such as temperature or smoke, exceeds some defined level, the state of the sensor becomes positive. At any point in time, the probability of a fire will be determined based on which of the sensors is positive or negative, as discussed previously. In reality, the detected value will be monitored over time, and different sensor variables may increase or decrease. By monitoring how these variables change, it is possible to get an understanding of the type of fire that may be developing. Different types of fire develop in different ways, some fires are characterised in their early development for example by much smoke, but little heat. By using such knowledge of fire 'fingerprints', more understanding of the fire type can be gained. Figure 9 illustrates how, at any point in time, the probability of each fire type can be determined based on the relative values of the sensors.

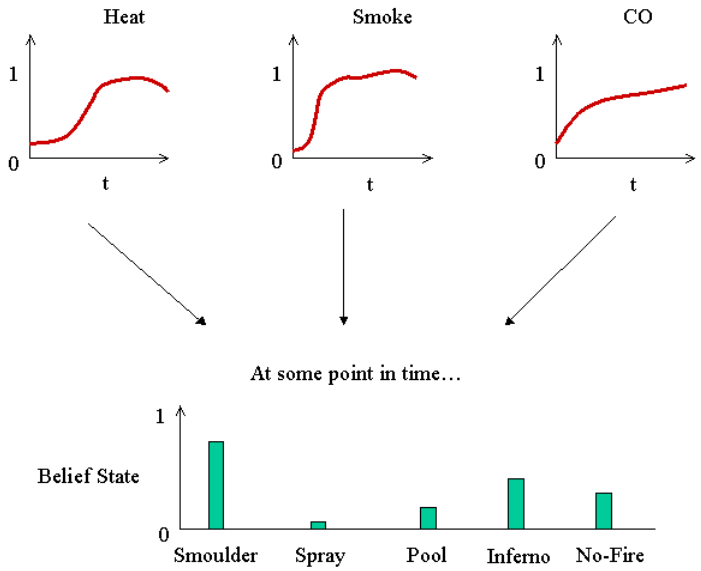

Figure 9 – A More Sophisticated Sensor Model

Introducing Time

Note that the network described earlier is static, that is it gives the belief state for the fire at an instant in time. As the fire detection system is a real-time system it must respond to changes over time. Although it could be possible to deal with this as a series of static snapshots, the network can actually use its previous knowledge of the environment to enhance its current belief. Dynamic Probabilistic Networks (DPNs) (ref. 3) provide a way of doing this where variables take on different values over time. The basic structure of a DPN is shown in figure 10. A network structure is repeated over regular time slices. The DPN can be thought to contain two models, the sensor model, formed by the arcs going to the sensor nodes, and the state evolution model, formed by the arcs going from one time slice to the next. The state evolution model quantifies how the network believes the state of the system changes over time.

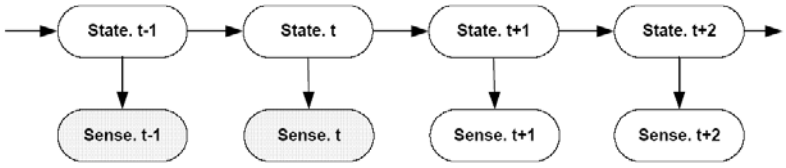

Figure 10 – DPN Structure

DPNs model the system as a partially observable Markov process, which means nodes can be not only connected to other nodes within the same time slice, but also to nodes in the time slices immediately preceding and following. The Markov property says that the future state of the system is independent of the past, given the present. This property means that the system does not need to maintain its history in order to predict the next state. The current belief state effectively captures the accumulated effect of previous observations. The shaded ovals in figure 10 represent the observations that are available at time t when predicting the state at time t+1. In the fire detection example, the network could use its current sensor observations, as well as its knowledge of the previous fire belief state, to determine the current fire type. This would be particularly useful where there were many fire types. The chance of having a fire of a particular type will vary depending on the previous fire type. The probabilities of different types would be dependent upon the way in which fires develop and propagate.

Network Structure

The structure of the BBN, that is the way in which the nodes are connected together, is often thought of as representing causal links between variables (for example the fire type in figure 1 has a causal effect on the sensor readings; the values of the sensor readings are directly influenced by the fire state). In fact the links between the nodes in a BBN do not necessarily represent causal relationships. It would be possible to represent the network in figure 1 using a different structure. This would require different CPTs for the nodes in order to correctly represent the different probabilistic relationships between the nodes. Although the links between the nodes would be different, the network would still provide the same information in terms of the probabilities for the nodes in the BBN. This is particularly important for networks which are machine learnt, that is BBNs which are constructed by a computer using data. For machine learnt networks the structure is unimportant as the network will be constructed such that it correctly represents the data used. For networks which are constructed by human experts, the structure of the network becomes much more important. It is important for the human to be able to understand the causal relationships between the nodes, such that the network can be verified. The structure may also represent other properties, such as conditional independence between variables, which need to be preserved. It is possible for a BBN to be constructed from both machine learnt elements, and elements developed using human expertise. Deciding which elements should be developed in which way depends upon the amount of data and expertise available, and also on how important structural properties of the network are.

Ensuring Safety

The fire detection system is relied upon to indicate the presence of a fire. As such it is a safety critical, or safety related, system. If a genuine fire is not detected quickly enough, then the fire fighting is delayed, potentially endangering human life and increasing asset damage. However, it is also important that false alarms are avoided, particularly if the fire fighting systems were automated, as fire fighting could cause extensive damage, as well as

endangering people in the vicinity. Although the consequences of a false alarm could, perhaps, be considered to be less severe than a delayed response, both cases are considered hazardous.

To ensure the safety of the system therefore, the chance of both non-detection of genuine fires, and of false alarms must be reduced to an acceptable level. If the fire detection system network from figure 1 is considered, the belief state of the fire type node is used to decide on the existence of a fire. An approach to ensure safety would therefore be to reduce the chance of incorrectly determining the presence of a fire to an acceptable level. As illustrated in figure 5, with all sensors showing false, the probability of a fire is 0.034%. Given this belief, it seems safe to conclude that a fire does not exist. Another interpretation of this belief state could be that a fire does in fact exist 3 in 10,000 times these observations are taken, a probability of failure on demand (if it were concluded there was no fire) of the order of 10^{-4}. From a safety perspective this is probably acceptable. Let us now consider figure 6. In this example one of the sensors is reporting a fire. The probability that a fire exists is 11.8%. Can it be safely concluded that a fire does not exist given this belief state? Now there is a probability of failure on demand of 10%. This is clearly unacceptable. Indeed, except in the very extreme belief states such as figure 5 it is hard to argue the safety of the system. Clearly this approach would be impossible to justify. If BBNs such as this are to be used successfully in safety critical roles then a different approach is needed. In the next section a possible different approach is considered.

Using BBNs for Decision Making

In order to ensure the safety of the fire suppression system, it is crucial that the correct decisions are taken on whether to suppress a fire or not. The correct decision will be
the one which leads to the safest outcome given the current belief state of the system. In this section an approach to decision making called Partially Observable Markov Decision Process (POMDP) is described. Initially a very simple example taken from reference 4 will be used to illustrate the approach, later, the way in which this approach may be applied to the safety of systems such as the fire detection system will be investigated.

The example that shall be used is referred to as the ``Tiger Problem". There are two doors, behind one door is a large and aggressive tiger, behind the other is a reward. If the door with the tiger is opened then there is a large penalty (being savaged). As well as opening one of the doors, there is also the option to listen in order to gain more information about the tiger's location. The act of listening also has a small penalty associated with it, to reflect the fact that the goal (receiving the reward) is being delayed. Listening for the tiger is not completely accurate, there is a chance that when a tiger is heard behind the left door, it is in fact behind the right (and vice versa). The example can be represented as a BBN with 2 nodes as shown in figure 11. Tiger left represents the belief state that the tiger is on the left (the belief state for tiger right is 1 minus this belief state). Heard left represents the probability of hearing the tiger on the left. The probability that a tiger on the left is heard on the left is 85%, and 15% that its heard on the right. Conversely for a tiger on the right.

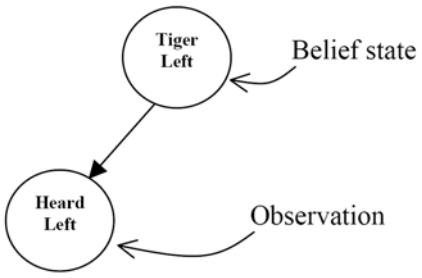

Figure 11 – BBN for Tiger Problem

Values can be assigned to the rewards and penalties to represent their relative weighting. The reward for opening the correct door is +10. The penalty for opening the door with the tiger behind it is -100. The cost of listening is -1. Without any bias, such as information from a listening observation, there is a 50% belief in the position of the tiger (it is equally likely to be behind either door). It is possible to calculate the expected reward of opening a door in such a situation. The result can be calculated by multiplying the belief state, by the value of the outcome resulting from that state. So in this case:

reward with b=0.5= ((prob. correct door) * (reward for correct door)) + ((prob. incorrect door) * (penalty for incorrect door)) = (0.5*10) + (0.5 * -100) = -45

The value of listening is -1, which is greater than the expected value of opening a door at random (-45). Therefore, the best thing to do in this situation is not to open a door but to listen, in order to gain more information. In fact it is only when you start with an initial belief state of at least 90% in the tiger's position that the first action should be to open a door (the value of doing so is greater than -1).

reward with b=0.9 = (0.9 * 10) + (0.1 * -100) = -1

It's possible to represent the optimal policy in a policy tree. The single step policy tree for the tiger problem is shown in figure 12. The nodes represent the policy that should be adopted (open the left door, open the right door, or listen), and the intervals below each node represent the range of belief state probabilities over which that policy dominates. It can be seen that the policy is to listen unless the belief state is at least 90% either way.

Figure 12 – Single Step Policy Tree

If a policy with more than one step is considered, it is possible to again calculate the values. Figure 13 shows a two step (t=2) policy tree. The second step is exactly as for the single step policy tree. The arcs from one time step to the next indicates the result that leads to the belief state in the next time step (TL is tiger observed left, TR is tiger observed right). The interesting thing to notice in figure 13 is that the first step is always to listen. It is always better in a two step strategy to listen at the first step such that a more informed choice is made at the second. It is only at t=4 that non-listen nodes appear again, as there are now belief states where the expected value from opening a door, is greater than that for listening.

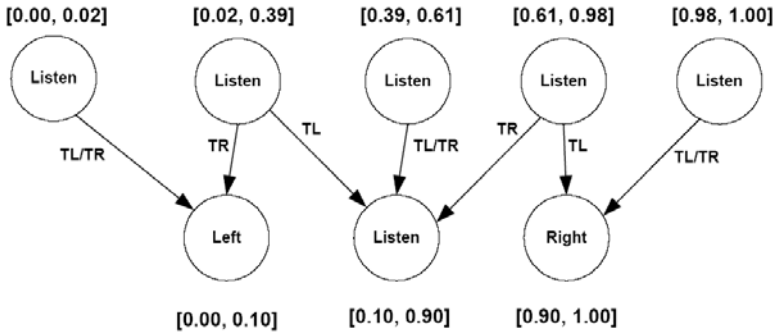

Figure 13 – Two Step Policy Tree

As the number of steps becomes large, the policy tree starts to have the same structure from one time step to the next. This first happens to appear in this tiger problem at t=56. It is possible then to redraw the policy tree such that the edges from one level are redrawn to itself as if it were the succeeding level. This results in a stationary policy referred to as a plan graph, which can be seen in figure 14. The key thing of note here is that the belief state no longer matters. The policy is dependant only on the current action node. Essentially the policy is to keep listening until the tiger is heard twice more on one side than on the other.

It should be noted that this plan graph is only true with the 85% probability of an observation being correct. If the observation is less accurate then the resulting plan graph is much larger, as it would be necessary to listen for longer

to be sufficiently confident to act. If the probability of hearing the tiger in the correct position reduces to 65%, then the plan graph becomes as shown in figure 15. This illustrates how the probabilities associated with the observations made, affect the policy, rather than the probability on the belief state node from the BBN.

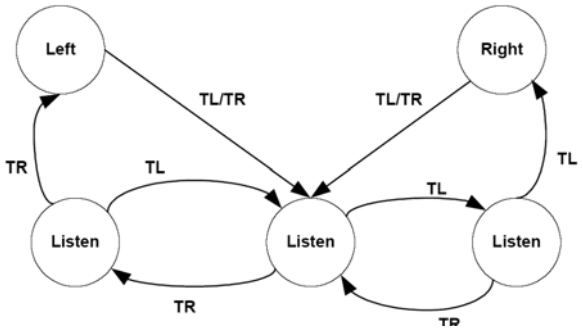

Figure 14 – Plan Graph for the Tiger Problem

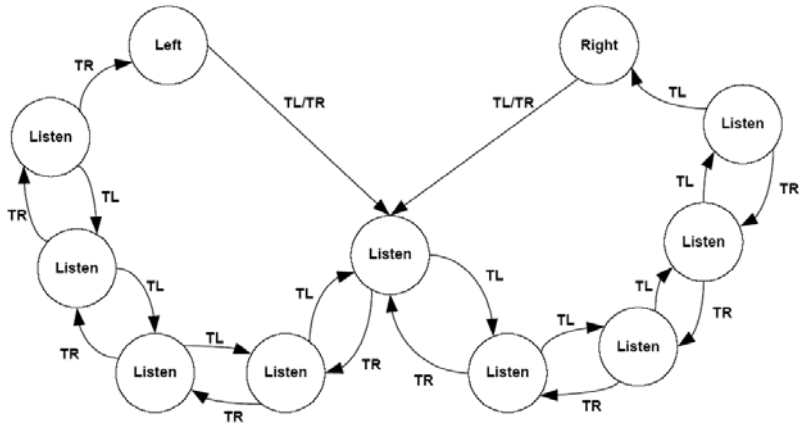

Figure 15 – Plan Graph for the Tiger Problem with Less Reliable Sensors

Applying POMDP to Safety Critical BBNs

Having seen how a POMDP can be applied to the simple tiger problem, there are clear analogies that can be made to the fire detection system discussed earlier in this paper. In both cases there is a belief state which is maintained with reference to observations made about the environment. In both systems the observations made are unreliable in some way. The fire detection system has a set of actions that may be performed; these are to initiate fire suppression, or to wait for further observations. The action of waiting for further observations is particularly important. Not all sensors may report a result at the same time. Some sensors may detect a fire earlier in its development that others, also some detectors may be positioned in closer proximity to the fire source. The actions will result in outcomes which can be given values, a reward for suppressing a fire and a penalty for not suppressing a fire, for unnecessary fire suppression, and waiting for further observations. One of the key challenges to adopting this approach would be to define the value functions such that they generate safe policies. There are reinforcement learning techniques available, such as the Q-learning algorithm (ref. 5), which may be used in the generations of value functions. Purely as an example, by recognising the relevant severity of the outcomes, the following values might be assigned:

Fire suppressed = 10
Unsuppressed fire = -100
Unnecessary fire suppression = -30
Await observations = -1

This information could then be used to generate policy trees, and hence plan graphs for the fire suppression system. The policies generated should ensure that the safest action is chosen for any given system state. The policy adopted would be related to the accuracy and reliability of the sensor information.

Safety Challenges

This document has attempted to highlight many of the issues associated with using BBNs as part of real-time safety critical systems. Many of these issues are crucial to the successful adoption of BBNs to safety critical roles. It has been seen how the belief state cannot give enough confidence in the state of the system to guarantee the system's safety. This has nothing to do with the correctness of the BBN, but instead with the ability to reason with confidence about probabilistic classifiers. By using a POMDP approach, the focus of the problem instead becomes ensuring that an accurate sensor model is produced and maintained. Although this is still a difficult and challenging problem, it is at least achievable.

A key element of any approach must be to integrate the safety aspects of the system, and the Bayesian network aspects. From this point of view, the starting point in developing safety critical BBNs must be the system hazards for the system in which the BBN is to be used. The development of the network must then ensure that these hazards are mitigated. In order to do this, each hazard must be related to a critical node in the BBN structure, and decision nodes relating to the hazard scenarios included in the network. For example, an unsuppressed fire hazard is related to the fire state node of the BBN in figure 1, and a suppression decision node would be included. Causal links between this critical node and others in the network can be assessed, and decisions made as to where machine learning would be appropriate. At this point the CPTs for the nodes have not been established, but a stable structure should be emerging. The decision node, whether implemented using a POMDP or some other mechanism, can be use to derive requirements on nodes in the network. For example a POMDP for fire suppression in the example in this paper can be used to derive safety requirements on the sensor nodes, which will be accuracy and reliability requirements. These safety requirements must be met by those nodes to ensure a safe policy is implemented.

Conclusions and Future Work

Here the main safety challenges, as identified in this paper, are highlighted, their resolution being future research requirements. The order of the issues is not intended to indicate any priority.

- Accurate initial CPTs are required for the sensors. It is this information which is the key to the accuracy of the network. These CPTs would also need to be validated in some way to ensure they truly represented the performance of the sensors. Learning algorithms, such as those developed in references 6 and 7, have been used to develop CPTs for some systems.
- The network should also learn from the performance of sensors, and use this information to ensure that the CPTs remain accurate. The use of sensor status nodes was discussed, as a way of handling degradation or failure of the sensors. The network must also take account of sensors which seem not to perform as described in their CPT, for example a sensor with large numbers of false positives must be considered less reliable, and the CPT should reflect this.
- Clearly the values assigned to outcomes is key to generating the correct policy for a given network. The generation of these values to unsafe outcomes needs to in some way encapsulate the risk associated with that outcome. The effectiveness of things like the Q-learning algorithm in generating values needs more investigation.
- This paper has deliberately focused on simple examples in order to extract the main issues. More complex networks will introduce more complex challenges, and issues of scalability will be crucial to the successful use of such systems.
- Both the structure, and the probabilities within the network must be validated to an extent where certification is achievable. For the Bayesian Automated Taxi system (ref. 8), a critical error rate below 1 in 10^8 seconds (i.e. equivalent to a good driver) is the target for the system, which they believe to be attainable.
- Certification will also require that an argument can be constructed that the network can be relied upon in a safety critical context, whether in a control, or an advisory role.

References

1. Norsys Software Corporation., Netica, Application for Belief Networks and Influence Diagrams. Vancouver, Canada, version 1.05, 1997.
2. J. Forbes et. al., The BATmobile: Towards a Bayesian Automated Taxi, International Joint Conference on Artificial Intelligence, 1995
3. T. Dean and K. Kanazawa., Probabilistic Temporal Reasoning. Proceeding of the Seventh National Conference on Artificial Intelligence, pages 524-528, 1988.
4. L. Kaelbling et. al., Planning and Acting in Partially Observable Stochastic Domains. Technical Report, Brown University, Providence, RI, 1997
5. Chris Watkins., Learning from Delayed Rewards. PhD Thesis, University of Cambridge, UK, 1989
6. J. Binder et. al., Adaptive Probabilistic Networks and Hidden Variables. Machine Learning, 1997
7. S. Russell et. al., Local Learning in Probabilistic Networks with Hidden Variables. Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995
8. J. Forbes et. al., Feasibility Study of Fully Automated Vehicles Using Decision-Theoretic Control. Tech. Report UCB-ITS-PRR-97-18, Computer Science Division, University of California, Berkeley, 1997

Biography

R. D. Hawkins, MSc., Research Associate, Department of Computer Science, University of York, York, YO10 5DD, UK, telephone - +44(0)1904 433385, facsimile - +44(0)1904 432708, e-mail – richard.hawkins@cs.york.ac.uk.

Richard Hawkins has been a Research Associate in the BAE SYSTEMS Dependable Computing Systems Centre at the University of York since November 2001. His research interests include the use of probabilistic systems, and object oriented techniques for safety critical systems. Before taking up his current role, he attained an MSc in Information Systems from the University of Liverpool and worked as a safety adviser for British Nuclear Fuels from 1997.

J. A. McDermid, PhD., Professor of Software Engineering, Department of Computer Science, University of York, York, YO10 5DD, UK, telephone - +44(0)1904 432726, facsimile - +44(0)1904 432708 e-mail – john.mcdermid@cs.york.ac.uk

John McDermid has been Professor of Software Engineering at the University of York since 1987 where he runs the high integrity systems engineering (HISE) research group. HISE studies a broad range of issues in systems, software and safety engineering, and works closely with the UK aerospace industry. Professor McDermid is the Director of the Rolls Royce funded University Technology Centre (UTC) in Systems and Software Engineering and the BAE SYSTEMS-funded Dependable Computing System Centre (DCSC). He is author or editor of 6 books, and has published about 250 papers.