Improved Methods for building large-scale Bayesian Networks Statement of Interest for the Third Bayesian Modelling Applications Workshop at UAI 2005

Martin Neil and Norman Fenton
Agena Ltd
and
Risk Assessment and Decision Analysis Research (RADAR) group,
Department of Computer Science,
Queen Mary, University of London

1. Background

Agena Ltd and the RADAR group have been applying Bayesian Networks (BNs) to risk assessment problems in a variety of problem domains for the last five years. Prominent areas in which we have applied BNs include:

- Predicting the risk of mid-air collisions between aircraft in UK airspace [17];
- Predicting software defects in complex consumer electronic devices [2,6,8,9];
- Evaluating the reliability and availability characteristics of military systems [15];
- Modelling the warranty return rates of electronic components [3];
- Modelling operational risk in financial institutions and predicting resulting losses. [14]

All of these examples involved building large-scale BN models for a real end-user. To support this work we have developed a general BN decision support tool called AgenaRisk [3]. In addition to this we have also developed a number of heuristics that have been successfully deployed and validated in these application areas. We would hesitate to claim these heuristics constitute a methodology, but are confident that they have provided, and will continue to provide, a productive and reproducible means for modelling uncertainty in risky domains involving limited or ambiguous data and the consequential requirement to accommodate expert judgement.

Our heuristics address two different, but universally applicable steps, encountered when using BNs:

- Deciding the conditional relations embodied in a Directed Acyclic Graph (DAG) — we address this using the idea of idioms.
- Completing the node probability tables (NPTs) — for particular classes of NPTs we use ranked nodes.

2. Idioms

When working with experts in a variety of domains a series of fundamental questions arose concerning the structure of the DAG:

- Which edge direction to choose? What was the cause and which the effect?

- Whether some of the statements they wished to make were actually uncertain and if not whether they could be represented in a BN?

- What level of granularity was needed when identifying nodes in the BN?
- Whether competing models/views could somehow be reconciled into one BN model at all?
- How could the notions of conditional and unconditional dependence be used to specify dependencies between nodes in a non-technical way?
- Whether we could use a "divide and conquer" approach to manage complexity?
- Could previously encountered BN patterns be reused in different domains?

We have identified a small number of, apparently, natural and reusable patterns in reasoning to help when building BNs [16]. We call these patterns "idioms". We use this term to refer to specific BN fragments that represent very generic types of uncertain reasoning. We are interested only in the graphical structure and not in any underlying probabilities and so for this reason an idiom is not a BN as such but simply the graphical part of one. We have found that using idioms speeds up the BN development process and leads to better quality BNs.

In our view fragments, as defined by Laskey and Mahoney [12], constitute smaller BN building blocks than idiom instantiations. Syntactically an idiom instantiation is a combination of fragments. However, we would argue that an idiom instance is a more cohesive entity than a fragment because the idiom from which it is derived has associated semantics. A fragment can be nothing more than a loose association of random variables that are meaningful to the expert, but the semantics of the associations within a fragment need to be defined anew each time a fragment is created. Thus, the use of fragments *only* does not lead to reuse at the level of reasoning, only at a domain specific level.

We claim that for constructing large BNs domain knowledge engineers find it easier to use idioms to construct their BN than following textbook examples or by explicitly examining different possible d-connection structures between nodes, under different evidence scenarios. This is because the d-connection properties required for particular types of reasoning are preserved by the idioms and emerge through their use. Also, because each idiom is suited to model particular types of reasoning, it is easier to compartmentalise the BN construction process.

3. Rank nodes

In the absence of hard data, we must rely on domain experts to provide, often subjective, judgements to inform the values used in NPTs. A key challenge is to construct relevant NPTs using the minimal amount of expert elicitation, recognising that it is rarely cost-effective to elicit *complete* sets of probability values. Historical methods for this, as used for labelled and Boolean nodes, include the so called Noisy-OR distribution and variants thereon such as the Noisy-MAX distribution [19].

We describe a simple, alternative, approach to building NPTs for a large class of commonly occurring nodes, which we have called ranked nodes [4]. Ranked nodes are given state values in ranked order ('very high' to 'very low') and we assume that they have metric properties, despite being ranked. This is a useful compromise to make in practice if we want to avoid overburdening experts with the effort of completing every probability value in a large NPT.

The approach has been partially automated and is thus accessible to all types of domain experts, including those with little statistical expertise. We have used rank nodes, in collaboration with domain experts to build large-scale realistic BN models that solve important problems in a realistic way.

We use rank nodes in two ways

1. To model causal relationships between many parent nodes and one child node. Here we use regression modelling to complete the probability table and assign weights proportional to reflect the correlation between each parent and the child.
2. To model "indicator" type relationships between many children and one parent. A Kalman filter analogy is used to model these types of relationships. Again, here, weights are applied to represent the correlation between the indicator, child, node and the indicated, parent node.

Indicator nodes operate in a similar way to "filter" nodes in a Kalman filter. In object radar tracking applications a Kalman filter operates by filtering out the noise from noisy radar observations in order to reveal the true position of the object in space. Here we can think of the indicators as providing noisy or imperfect observations and the parent node as the true, perhaps unobservable, value awaiting estimation. In a Kalman filter we wish to condition our estimate for the "true" value on the data to hand from each of our "indicator" nodes assuming each indicator is Gaussian distributed.

We use rank nodes to model NPTs as symmetric, truncated conditional distributions using a doubly Truncated Normal distribution. The mean value of the distribution is then modelled as a simple arithmetic average, weighted average, min. or max. function. The variance parameter is then used as a measure of credibility or "importance" where higher variance indicates lower credibility and vice versa.

In our experience we have found that experts are quite comfortable with the use of ranked nodes. In fact, in some domains, they actually expect to interpolate/extrapolate values in an NPT when faced with the burden of completing a large number of columns. Often we have seen experts quickly invent on-the-fly heuristics to collapse the NPT to a small set of canonical rules that allow them to model the whole NPT using these few rules. In these situations we can exploit the rank nodes idea quickly and effectively.

In practice we support the elicitation of the weights by collaboratively modelling "what-if?" combinations of parent state configurations and asking the experts to describe the conditional distribution terms of shape, central tendency and variation. For instance for a large NPT we might sample a few combinations of parent states, deduce some weights (either by employing Bayesian learning using a hierarchical model or fitting "by eye"), demonstrate the fitted NPT back to the expert and then repeat this process until both parties are happy with the result. The end result is a verified NPT reflects the expert(s) beliefs about the relationship.

Some caution is required when following this process, for a number of reasons. Firstly, clearly not all NPTs can be modelled in this way. The nuanced interaction between parent nodes might wrongly be ignored when modelling NPTs thus giving result to considerable inaccuracy. Therefore great care needs to be taken to present combinations of causes to experts that will prompt them to admit to differences in effect, should such differences exist in their mind or in actuality. Secondly, we must stress that a verified NPT that matches the expert's beliefs is not necessarily a valid NPT and when combined with other idioms/fragments in the whole BN additional reconfiguration and recalibration might be necessary.

4. Summary

One of the key challenges for BN researchers is to address the difficulty (for both experts and non-experts) of building and maintaining (as opposed to just executing) large-scale models necessary to solve real-world problems. In meeting this challenge on real commercial projects we had to make it possible for users with minimal statistical knowledge to build and edit large-scale models for a range of application domains. As in other complex knowledge engineering tasks, we have found that heuristics and patterns for reuse are critical to the effectiveness of any knowledge engineering task. Our use of idoms (as patterns) and ranked nodes has proven effective in helping to build a large-class of models that are now in everyday use.

References

1. Druzdzel MK and van der Gaag LC, "`Building Probabilistic Networks: Where Do the Numbers Come From?'", IEEE Transactions on Knowledge and Data Engineering, 12(4):481-486, 2000.
2. Fenton N, Krause P, Neil M, "Probabilistic Modelling for Software Quality Control", Journal of Applied Non-Classical Logics 12(2), 173-188, 2002
3. Fenton NE and Neil M, "Combining evidence in risk analysis using Bayesian Networks", Safety Critical Systems Club Newsletter 13 (4) Sept 2004
4. Fenton NE and Neil M, "Ranked nodes: A simple and effective way to model qualitative judgements in large-scale Bayesian Nets", submitted Knowledge Based Systems Feb 2005
5. Fenton NE and Neil M, "The Jury Observation Fallacy and the use of Bayesian Networks to present Probabilistic Legal Arguments", Mathematics Today (Bulletin of the IMA, 36(6)), 180-187, 2000.
6. Fenton NE, Krause P, Neil M, "A Probabilistic Model for Software Defect Prediction", accepted for publication IEEE Trans Software Eng, Sept 2001
7. Fenton NE, Krause P, Neil M, "Software Measurement: Uncertainty and Causal Modelling", IEEE Software 10(4), 116-122, 2002
8. Fenton NE, Marsh W, Neil M, Cates P, Forey S, Tailor M, "Making Resource Decisions for Software Projects", 26th International Conference on Software Engineering (ICSE2004) May 2004, Edinburgh, United Kingdom. IEEE Computer Society, ISBN 0-7695-2163-0, pp. 397-406, 2004.
9. Fenton NE, Neil M, Marsh W, Krause P, Mishra R, "Predicting Software Defects in Varying Development Lifecycles using Bayesian Nets", submitted to ESEC 2005
10. Joseph A, Fenton NE, Neil M, "Predicting football results using Bayesian Nets and other Machine Learning Techniques", 2005
11. Koller D and Pfeffer A, "Object-Oriented Bayesian Networks", Proceedings of the 13th Annual Conference on Uncertainty in AI (UAI), Providence, Rhode Island, August 1997, pages 302--313, 1997.
12. Laskey KB and Mahoney S, "Network fragments: representing knowledge for constructing probabilistic model networks", 13 Annual Conference on Uncertainty in AI, http://site.gmu.edu/~klaskey/lectures.html, 1998.
13. Laskey, KB. Mahoney, S.M, "Network engineering for agile belief network models", IEEE Transactions on Knowledge and Data Engineering, 12 (4), 487-498, 2000.
14. Neil M, Fenton N, Tailor M, "Using Bayesian Networks to model Expected and Unexpected Operational Losses", 2005 accepted for publication, Risk Analysis: An International Journal
15. Neil M, Fenton N, Forey S and Harris R, "Using Bayesian Belief Networks to Predict the Reliability of Military Vehicles", IEE Computing and Control Engineering J 12(1), 11-20, 2001.
16. Neil M, Fenton NE, Nielsen L, "Building Large-scale Bayesian Networks", The Knowledge Engineering Review, 15(3), 257-284, 2000.

17. Neil M. Malcolm B. and Shaw R. “Modelling an Air Traffic Control Environment Using Bayesian Belief Networks”. 21st International System Safety Conference, August 4 - 8, 2003, Ottawa, Ontario, Canada.
18. Neil M., Krause P., Fenton N. Software Quality Prediction Using Bayesian Networks in Software Engineering with Computational Intelligence, (edited by Khoshgoftaar T. M). The Kluwer International Series in Engineering and Computer Science, Volume 731, 2003.
19. Takikawa M. and D'Ambrosio B. Multiplicative Factorization of Noisy-Max. Proceedings of the Uncertainty in AI conference ,1999.
20. van der Gaag LC, Renooij S, Witteveen CLM, Aleman BMP, Taal BG, "Probabilities for a Probabilistic Network: A Case-study in Oesophageal Carcinoma", University of Utrecht, UU-CS-2001-01, January, 2001.

Biographies

Martin Neil

Martin Neil is a part-time Reader in the Computer Science Department of Queen Mary, University of London. He holds a degree in 'Mathematics for Business Analysis' from Glasgow Caledonian University and a PhD in 'Statistical Analysis of Software Metrics' jointly from South Bank University and Strathclyde University. From 1996 - 1999 he spent four years at the Centre for Software Reliability, City University (London) and from 1992 - 1995 worked for Lloyd's Register and JP Morgan. His interests cover applications and theory in Bayesian probability to software quality, system dependability and operational risk in finance. Martin is a Charted Engineer.

Martin is also the Chief Technology Officer at Agena. Agena provides software products to solve complex risk assessment and decision problems: www.agenarisk.com (Contact: martin@agena.co.uk).

Norman Fenton

Norman Fenton is Professor of Computing at Queen Mary, London University, and also CEO of Agena, a company that specialises in risk assessment for critical systems. He is a Chartered Engineer and Chartered Mathematician. Norman's books and publications on software metrics, formal methods, and risk analysis are widely known in the software engineering community. His recent work with colleague Martin Neil has focused on the use of Bayesian Nets (BNs) for risk assessment in a wide range of application domains such as vehicle reliability, embedded software, transport systems, and financial services. In support of this, Agena has been building BN-based decision support systems for a range of major clients.