Things to know about Bayesian networks

Decisions under uncertainty, part 2

Bayesian networks help us model and understand the many variables that inform our decision-making processes. **Anthony C. Constantinou** and **Norman Fenton** explain how they work, how they are built and the pitfalls to avoid along the way

We constantly make decisions – some routine, such as what to wear, some more complex and important, such as choosing where to live and work. The more complex a decision, the less likely we are to have all the information we need to make the best possible choice. Because incomplete information requires us to reason with probabilities and risk, people (even experts) often arrive at non-optimal decisions.

More complex decisions are usually based on a host of factors or variables. For example, imagine you are the owner of a top-flight football (soccer) team and you must decide before the start of each season how much money to invest in new players. There are many factors to consider: the likely income from sales of unwanted players; the relative net spending of other teams; and the possible negative impact on team performance of making too many personnel changes at once. We can map the decision the team owner needs to make, and all the different variables, using a Bayesian network (BN). This is a graphical model that captures the relationship between variables under causal or influential assumptions.

In this article, we provide an overview of BNs and the kind of assumptions required to build useful networks for complex decision-making. We highlight the need to fuse data with expert knowledge, and we describe the challenges in doing so. Finally, we explain why, for fully optimised decision-making, extended versions of BNs, called Bayesian decision networks, are required.

The basics

A BN is a diagram which uses arrows ("directed arcs") to show how various factors – represented by elliptical nodes – influence one another. Each node comes with its own probability table, known as a conditional probability table (CPT), reflecting the chances of various outcomes resulting from the different influences directly affecting it. Figure 1

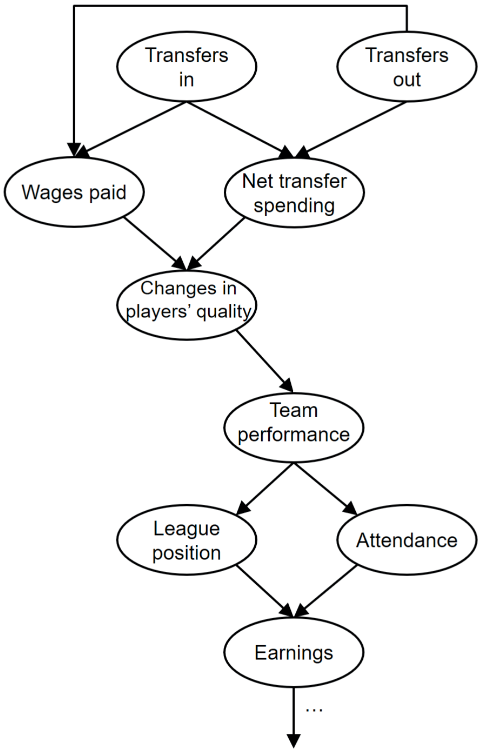

FIGURE 1 A fragment of a Bayesian network model for a football team owner, showing the variables related to team performance.1 Elliptical nodes are the factors or variables; arrows are the directed arcs showing how factors influence one another.

Anthony C. Constantinou is a lecturer in machine learning and data mining at Queen Mary University of London.

Norman Fenton is a professor in the School of Electronic Engineering and Computer Science at Queen Mary University of London, where he is also director of the Risk Information Management research group.

illustrates part of a BN model which provided long-term predictions for football team performance.[1]

Once the graphical structure and CPTs of a network have been defined, there are standard algorithms that compute the states of the unknown variables based on the states of the known variables in the model. We can enter as many or as few actual observations as we have available, and the algorithm will update the probability distribution of each of the unobserved variables.

A reason why BNs are so powerful is that they can perform both predictive and diagnostic inference. For example, we can predict a team's *league position* for a given value (observation) of *team performance*, but we can also enter a required state of *league position* as an observation to examine what level of *team performance* could explain that observation. These standard algorithms are called "Bayesian propagation" algorithms because they rely on Bayes' theorem, in which the probability of an unknown variable is updated after evidence relevant to that variable is observed. In a BN, Bayesian probability inference is driven by the three causal classes illustrated in Figure 2. Specifically:

1. **Causal chain:** This describes variables that have a knock-on effect on each other. For example, *changes in players' quality* impacts *team performance* which impacts l*eague position*. This means that *league position* is independent of *changes in players' quality* once we know *team performance*.
2. **Common effect:** This is where two different variables, such as *transfers in* and *transfers out*, both impact a third variable, such as *net transfer spending*. This means that *transfers out* is dependent on *transfers in* once we know *net transfer spending*.
3. **Common cause:** This is where two different variables, such as *league position* and *attendance*, are impacted by the same variable, such as *team performance*. This means that *attendance* is independent of *league position* once we know *team performance*.

These causal assumptions are also vital if we wish to reason about an intervention. Figure 3 shows two different representations of the relationships between four of the variables used in the BN model of Figure 1. If we rely on data alone to determine associations between variables, we would arrive at the model shown in (a), which is *not* a BN because it does not capture the direction of influence between factors. In contrast to model (a), model (b) indicates that an intervention on *attendance* will have no effect on other parts of the model, whereas an intervention on *changes in players' quality* will have an effect on all of the model variables. In model (a), the association between *league position* and *attendance* comes via the common cause, *team performance*. The causal assumptions established by model (b) allow us to simulate the effect of interventions, actions, or decisions and hence, in contrast to model (a), enable us to move from mere prediction to risk management.

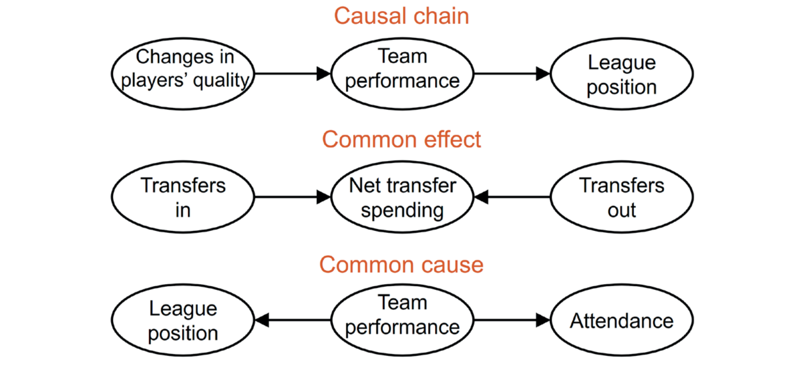

FIGURE 2 The three causal classes that produce Bayesian networks.

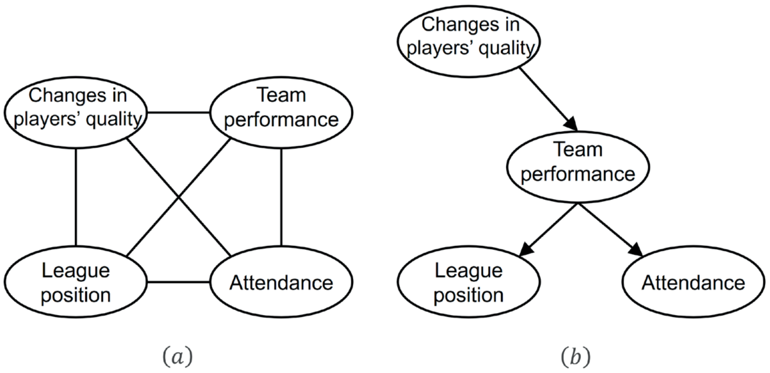

FIGURE 3 Models (a) and (b), under respective representations of association and causation.

Scientific research is heavily driven by interest in discovering, assessing, and modelling cause-and-effect relationships as guides for action. This means they require models more like (b) than (a). Although the distinction between association and causation is nowadays well understood, what has changed is mostly the way the results are stated, rather than the way the results are generated. Consequently, too often, important conclusions and recommendations are based on models similar to (a), and we believe this is a problem.

Building a Bayesian network

Constructing a BN involves determining both its structure and CPTs. We can do this by eliciting knowledge from domain experts (the knowledge-based approach), learning from data (the data-driven approach), or a combination of the two (information fusion). For the football model, we used both data and knowledge. Specifically, knowledge was used to construct the influential structure of the model, whereas data were used to learn the CPTs of the variables that make up the model.

In general, algorithms that deduce BNs from data (called "machine learning algorithms") can be classified as either "search-and-score" or "constraint-based" (or "hybrid", which is simply a combination of the two). In the search-and-score approach, we search for different structures and score them, using one or more scoring functions, based on how well the fitting distributions agree with the empirical distributions.

This approach typically aims to maximise predictive accuracy associated with targeted variables of interest. In the constraint-based approach, we check for causal conditions between variables in sets of triples in order to discover causal relationships by performing what are called "conditional independence" tests. For example, we can reconstruct the BN fragment of Figure 3 by discovering that while *changes in players' quality* and *league position* are predictive of each other, the association is eliminated conditional on *team performance*.

Issues with structure learning

Many of the proposed structure learning algorithms are assessed with simulated data, which are generated using hypothetical models that are assumed to represent reality, or "ground truth". Simulated data are then taken as input by an algorithm in an attempt to reconstruct the hypothetical model that was initially used to generate the input data. Figure 4 illustrates this process. Each algorithm is then judged both on its speed and accuracy in terms of how well the learnt model resembles the hypothetical model.

While structure learning algorithms tend to perform well when tested with simulated data, this level of performance is not repeated when the input data set represents observations from the real world. This is because of four main challenges:

1. **Missing data points:** Real-world data sets are often incomplete, yet many of the proposed structure learning algorithms require, and are assessed with, complete data sets. While some algorithms accept incomplete data sets as input, they tend to produce inadequate models.
2. **The need for really big data:** The accuracy of the learnt model depends on the available data. The number of possible models that can be discovered increases with the states and the number of variables that form the input data set. High-dimensional data sets, and resulting complex models, require extremely large volumes of data to learn a reasonably accurate BN. However, for many critical real-world problems we often have data for a large number of variables, but with (relatively) insufficient sample size. For example, the BN model in Figure 1 is based on just 20 samples (teams) for each of the 15 Premier League seasons between 2000 and 2015. Moreover, even when there are plenty of data, there is still the issue of that data potentially being biased, and thus likely to be precise but occasionally inaccurate.
3. **Latent confounders:** These are variables which are missing and which may have a major impact in explaining observed data. In our football model, the *squad stability* factor was missing from the data and, had it not been incorporated by expert judgement, as we later illustrate in Figure 5, it would have been a latent confounder. Structure learning algorithms rarely account for such variables and their resulting impact on what can and cannot be discovered. In the real world, latent confounders are impossible to avoid, simply because they are often unknown unknowns.
4. **Data quality:** To be able to learn the "correct" causal network, we require the "correct" data. Simulated data satisfy this requirement, since they are generated based on clearly defined models that are assumed to represent the ground truth. However, real-world observations rarely adhere to causal representation in the same way simulated data do. Because of this, results from simulated data tell us very little about the extent to which modelling assumptions hold true for real-world applications.

In addition, there is no agreed evaluation process to determine which algorithm is "best". Different evaluation methods often lead to inconsistencies, whereby one evaluator determines algorithm A to be superior to algorithm B, and vice versa. Moreover, there is a risk of erroneously rejecting a good algorithm while accepting a poor one. In practice, we never know what the ground-truth model really is and so require different evaluation procedures when applying structure learning to a problem in the real world.

In domains such as bioinformatics, applying structure learning algorithms to large data sets can reveal new insights that would otherwise remain unknown. But structure learning algorithms are less effective in areas where domain experts have knowledge about the underlying mechanisms of the problem. Incorporating knowledge into the structure learning process inevitably leads to better-quality models. Such knowledge comes in the form of constraints in terms of what can and cannot be discovered.

One constraint is temporal order, whereby event *B* occurs after *A*, and hence *B* cannot influence *A* (for example, *changes in players' quality* occurs first, so cannot be influenced by future *team performance*). We may also specify constraints about direct relation (for example, indicating that there is an arc between *attendance* and *earnings*, with or without specifying the direction of the arc). It is also possible to constrain the graph structure by specifying a simplified "best-guess" model with a metric that forces the algorithm to assign higher scores to models closer to the best guess. For example, since the structure of the BN model in Figure 1 is based on knowledge, it can serve as an initial best-guess model for a structure learning algorithm. An important benefit of such constraints is that they help to reduce the search space significantly, which relaxes some of the structure learning issues discussed earlier.

Information fusion

When the historical data fail to capture all of the key factors of interest, there are two options: either ignore the missing factors, or incorporate them as knowledge-based factors.

For example, the data used to construct the BN model in Figure 1 indicates that teams which increase wages and net transfer spending faster than adversaries improve their performance in the league, on average. However, what the data fail to capture, but for which we have knowledge, is that extreme increased spending does not necessarily translate immediately into improved performance, partly due to some form of team instability caused by multiple substantial

Acknowledgement
The authors acknowledge financial support by the European Research Council (ERC) project ERC-2013-AdG339182-BAYES_KNOWLEDGE, and the Engineering and Physical Sciences Research Council (EPSRC) project EP/P009964/1-PAMBAYESIAN.

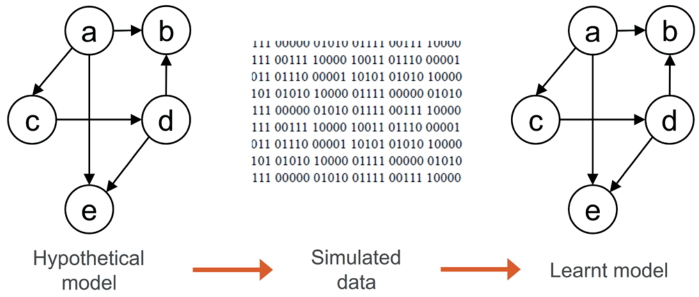

FIGURE 4 An illustration of structure learning with simulated data.

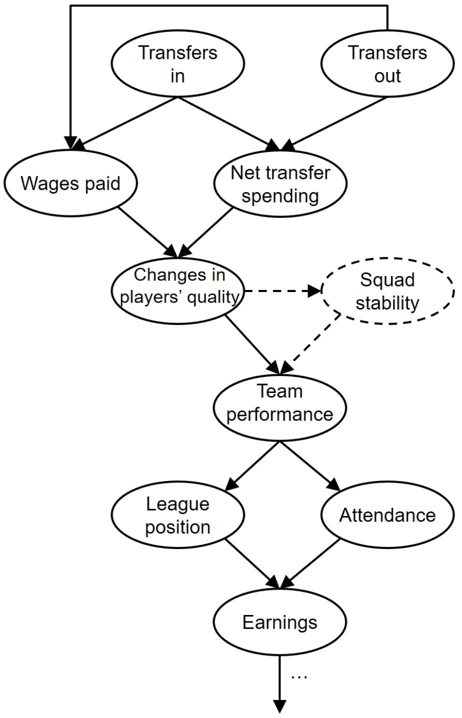

FIGURE 5 The Bayesian network model from Figure 1, now augmented with the knowledge-based missing explanatory factor, *squad stability*.

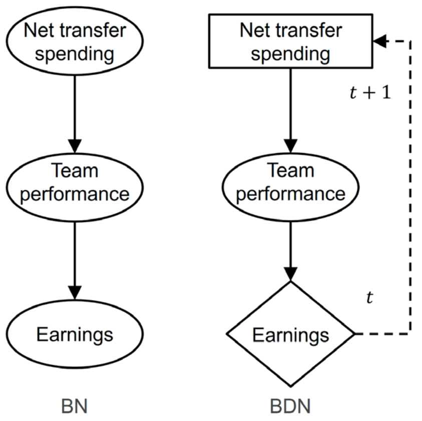

FIGURE 6 A Bayesian network with a (possible) corresponding Bayesian decision network.

changes in players. This may have been the case for Manchester City after it was bought by the Abu Dhabi United Group in 2008: the new owners purchased many high-profile players to improve the squad, but the team finished in a lower position with fewer points than the previous season. This missing factor – *squad stability* – is captured as a knowledge-based variable in the revised football model in Figure 5, indicated with a dashed node and dashed arcs.

In the absence of such explanatory factors, we run the risk of deducing from data alone the generalised expectations, which may negatively influence optimal decision-making. However, knowledge about missing explanatory factors enables us to question such inferences and improve the structure of a BN model and improve accuracy.[1]

But how do we properly fuse knowledge with data? It is vital to acknowledge that the statistical outcomes of data-driven variables are already influenced by the causes an expert might identify as variables missing from the data set. As a result, the incentive is to incorporate knowledge-based variables so that they do not influence data-driven expectations, as long as the knowledge-based factors remain unobserved within the model.[2]

In the football example in Figure 5, this means that the incorporated variable, *squad stability*, preserves the expectation of *team performance*, as long as the knowledge-based variable remains unobserved. Increased spending continues to correlate positively with team performance in the same way, but the augmented model explains the correlation in the knowledge-based variable and hence reduces the risk of flawed interventions.

Bayesian decision networks

A BN model provides an effective graphical representation of a problem and can be used for multiple types of complex inference. Despite these benefits, a BN model alone is incapable of determining the optimal decision pathways of a problem. For example, we may want to determine the optimal *net transfer spending* to achieve certain improvements in *team performance*. To achieve this, a BN needs to be extended to a Bayesian decision network (BDN), also known as an influence diagram. A BDN contains additional types of nodes and arcs, as illustrated in Figure 6.

In a BN, all nodes are considered uncertain "chance nodes". But in a BDN, if a node corresponds to a decision to be made we distinguish it as a "decision node" (drawn as a rectangle). We also introduce "utility nodes" (drawn as diamonds), which are targeted for maximising or minimising a particular outcome of interest, and "information arcs" (drawn as dashed arcs) entering decision nodes, indicating that the decision is determined by information retrieved from parent nodes. In contrast to normal BN arcs, information arcs only pass information forward.

Specifying a BDN inevitably requires expert knowledge since we need to specify the decision options available to the decision-maker, and the utilities we seek to minimise or maximise. In fact, there are certain structural rules we need to follow when transforming a BN into a BDN, such as ensuring that only informational arcs enter a decision node.[3,4] Figure 6 illustrates how a BN fragment may look when modified into a BDN, depending on which variables we define as decisions and utilities. Note that the example in Figure 6 also illustrates how earnings will determine next season's decision on net transfer spending. This process enables temporal analysis from a past BN to a future BN, and models extended towards this kind of analysis are called dynamic Bayesian (decision) networks.

Costs and benefits

With the wide availability of software that makes it easy to build and run models efficiently, there has been an explosion of interest in BNs for solving complex decision problems under uncertainty. In addition to the examples in this article, BNs have been used to evaluate the probative value of complex forensic evidence, and to provide more accurate diagnoses in medicine and more accurate predictions of financial risk. However, for fully optimised decision-making we require BDNs, which enable us to maximise or minimise different outcomes of interest – whether these involve the most cost-effective route, maximum impact, minimum risk, or an equilibrium between the two.

The benefits of BDNs, however, come at a cost of significant effort due to the high levels of manual model construction they currently require. This is because we need to establish the various decision support requirements and their associated cost and benefits, and then incorporate them into the model and appropriately combine them with data. For example, we may want to determine the optimal treatment combination in a particular sequence, to control unexplained symptoms or cure a disease, while at the same time taking care to minimise the risk of potential known and unknown side effects.

While some manual knowledge-based contribution is inevitable when building decision models, future research promises improvements in the ways we establish and model relationships (causal or other) between real-world factors, to allow for faster and more accurate optimal decision-making solutions under uncertainty. ■

References

1. Constantinou, A. C. and Fenton, N. (2017). Towards smart-data: Improving predictive accuracy in long-term football team performance. *Knowledge-Based Systems*, **124**, 93–104.

2. Constantinou, A. C., Fenton, N. and Neil, M. (2016). Integrating expert knowledge with data in Bayesian networks: Preserving data-driven expectations when the expert variables remain unobserved. *Expert Systems with Applications*, **56**, 197–208.

3. Hagmayer, Y., Sloman, S. A., Lagnado, D. A. and Waldmann, M. R. (2007). Causal reasoning through intervention. In A. Gopnik and L. Schulz (eds), *Causal Learning: Psychology, Philosophy and Computation*. Oxford: Oxford University Press.

4. Constantinou, A. C., Fenton, N., Marsh, W. and Radlinski, L. (2016). From complex questionnaire and interviewing data to intelligent Bayesian network models for medical decision support. *Artificial Intelligence in Medicine*, **67**, 75–93.

nfx702/Bigstock.com