



Bayesian Belief Networks

Abstract This chapter overviews Bayesian Belief Networks, an increasingly popular method for developing and analysing probabilistic causal models. We go into some detail to develop an accessible and clear explanation of what Bayesian Belief Networks are and how you can use them. We consider their strengths and weaknesses, outline a brief history of the method, and provide guidance on useful resources and getting started, including an overview of available software.

Keywords Bayesian Belief Networks • Bayesian networks

Bayesian Belief Networks (BBNs) were a real no-brainer to include in this book. Though they have been around for a little while now (30 years or so), we have seen a growing interest and excitement about them in a range of different fields, many concerned with social and policy questions. The roots of BBNs are in some of the more technical academic fields, such as computer science and statistics, but recognition of the value they can deliver has spread to many domains. There is a range of software options (more on these at the end of the chapter) to help you use BBNs, which mean that you don't need to have a deep understanding of the mathematics behind them. However, importantly, these don't give you 'too much power' without understanding, to risk doing genuinely inappropriate or misleading analysis. BBNs also fill an important niche in the landscape of different methods we explore in this book. They give us a method which

has some of the best bits of quantification, allowing us to ‘put numbers on things’ while also incorporating uncertainty in a meaningful way.

The term ‘Bayesian networks’ was coined by Judea Pearl in the late 1980s. He is an interesting and vocal character (his Twitter account is well worth a follow for the illuminating debates he instigates), and his body of work on causal inference has had a growing influence in recent years, most notably in economics, quantitative social science, and social data science. If you find this chapter interesting, and are looking for something more formal, we recommend also doing some reading around directed acyclic graphs (DAGs) and their use in causal inference.

Because of BBN’s roots in computer science, there is a large literature on their use based around the idea that we can use learning algorithms to generate their networks (or maps) directly from data. In common with Fuzzy Cognitive Mapping, there is a parallel literature on developing them in more participatory modes, from stakeholder knowledge. We will focus more on the latter in this chapter, though we will touch on the concepts behind developing BBNs, and other system map types, from data in Chap. 9.

As in all the methods chapters in this book, we will now give as clear and accessible a description of what BBNs are and how to do them, as we can muster. We will avoid the use of mathematical notation to do this, there are other introductions which cover the maths well (e.g. Neapolitan & Jiang, 2016). We will then explore some common issues and tricks of the trade for using them, try to pin down what they are good and bad at, before outlining a brief history of the method and pointing to some key resources for getting started yourself.

WHAT ARE BAYESIAN BELIEF NETWORKS?

BBNs, as with all networks, are made up of nodes (which here, represent variables, factors, or outcomes in a system) and edges (which represent the causal relations between these nodes). So far, so familiar, and like many of the methods in this book. What sets BBNs apart is that each node has some defined states (e.g. on or off, high or low, present or not present) and some associated likelihoods of being in each of those states. These likelihoods are based, in probabilistic fashion, on the states of the nodes that they are connected to, that is, the nodes from which they have arrows going into them. In the language of probability, nodes are ‘conditionally dependent’ on the states of the nodes that they have a causal relationship

with. So, the BBN is the network and the collection of conditional probabilities (usually shown in simple tables or plots annotated onto a network diagram), denoting the likelihood of nodes taking different states. The last key point to mention is that BBNs are acyclic; that is, they do not have any cycles or feedbacks, and the arrows must flow all in one direction. This is an important distinction between other methods which do have cycles, such as Causal Loop Diagrams and Participatory System Maps.

It is useful to make this more tangible quickly, so let's look at an example. Figure 7.1 shows a simple BBN of the effects of 'rainfall' and 'forest cover' in a river catchment, and the links through to some different outcomes, such as 'angling potential' and 'farmer income'. This is a simple BBN, you can see the acyclic structure, and focus on outcomes we or others might care about.

The same BBN, but this time with the states of each node shown along with the probability distribution for them is shown in Fig. 7.2. This allows us to see, for example, if 'rainfall' and 'forest cover' are both high, the likelihood of all the other nodes taking specific values.

We could explore different scenarios by setting the states of 'rainfall' and 'forest cover' differently and seeing how this affects the rest of the map. This is a common way of using BBNs, setting certain node states given our observations (or hypothetical scenarios we are interested in), and seeing what this implies about the probability of states in other nodes. With 'rainfall' and 'forest cover', we are setting values at the 'top' of the network (sometimes referred to as 'root nodes' or 'parent nodes', i.e. with

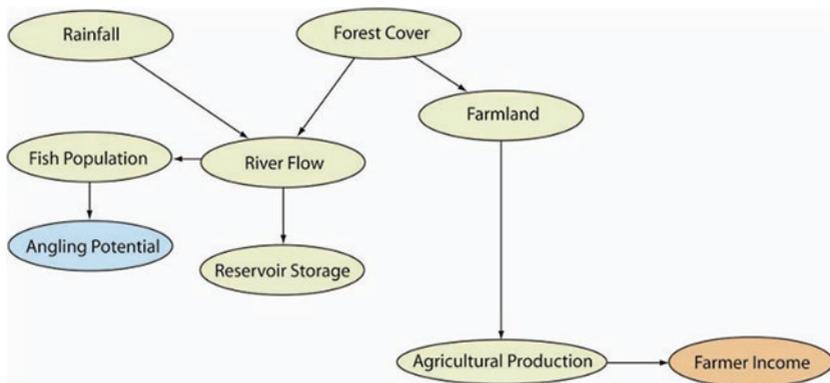


Fig. 7.1 An example simple BBN. (Source: Bromley (2005))

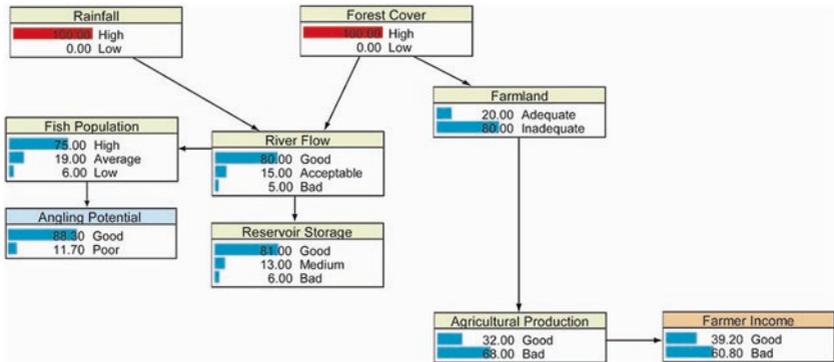


Fig. 7.2 An example BBN, now with nodes states and probability distributions. (Source: Bromley (2005))

Table 7.1 An example conditional probability table based on the reservoir storage node in the BBN in Figs. 7.1 and 7.2

		River flow		
		Good	Acceptable	Bad
Reservoir storage	Good	0.9	0.6	0
	Medium	0.1	0.3	0.1
	Bad	0	0.1	0.9

Source: Bromley (2005)

no nodes going into them) and looking causally ‘down’, but it can be done the other way round too; setting the states of outcomes (sometimes referred to as ‘leaf nodes’ or ‘child nodes’) and looking ‘up’ the network to see what might have contributed to that outcome. These are the two main types of insight the analysis of BBNs can provide: (i) assessing the probability of achieving outcomes, and (ii) quantifying the impacts on outcomes of changes elsewhere in the system. As with all the methods in the book, there is also huge potential value in the process of building a BBN, to generate discussion and learning about the topic.

These examples help us get a quick sense of what BBNs are about, but they are focused on the ‘results’ of the BBN; this is what is shown in the node tables. What is not present are the conditional probability tables that underpin these outputs. Table 7.1 shows what one of these tables might look like for one of the factors in this BBN, ‘reservoir storage’. It

shows different states of the parent node of ‘reservoir storage’, which is ‘river flow’, and the resulting probabilities of ‘reservoir storage’ taking each of its states. The numbers indicate the probability that ‘reservoir storage’ will take its values in the second column (i.e. if ‘river flow’ is good, then there is a 90% chance ‘reservoir storage’ is good, and 10% chance it is medium).

BBNs have some well-known and often-criticised constraints. First, they are acyclic; that is, there cannot be any feedback loops, of any length, in the network. This constraint is imposed for the calculations to work. In complex systems, it is rare for there to be no feedback loops; where there are feedback loops, these are often powerful drivers of dynamics in the system. It is possible to partially represent feedback loops by including multiple nodes for the same thing, but for different time points (we show an example of this below). BBNs that use this approach are often called ‘dynamic’ BBNs. A second constraint on BBNs developed with expert input is that most nodes cannot have more than two or three parent nodes (i.e. incoming connections) and nodes should not have more than a handful of states. This constraint is normally imposed so that the conditional probability tables, which are a key component of what is elicited from stakeholders to build the BBN, do not become unworkably large. By way of illustration, imagine a node with two states, and two parents each with two states themselves, this will require a 4×4 table. However, a node with three states, and three parents, each with three states themselves will need a 6×27 table. Imagine filling that in with stakeholders, cell by cell, with potentially important discussions at each step, and doing this for every node in the map. Combined, these two constraints mean that the underlying network in a BBN tends to end up being a relatively simplified model of reality compared to some of the other systems mapping methods in this book. This is not necessarily a problem, but it is a constraint we should be aware of.

These constraints often invoke ire from researchers and practitioners who want to represent whole systems and take a complex systems world-view (including from us in the past!). However, one of the typically misunderstood, or simply missed, nuances with BBN is that the use of conditional probabilities means that we can still capture some elements of the wider system in the analysis and discussion around constructing maps, even if they are not in the network explicitly. To demonstrate, consider Table 7.2. Here, we can see the probability of an outcome occurring given the state of two interventions. Even when we have both interventions ‘on’

Table 7.2 Simple hypothetical conditional probability table for two interventions and an outcome

<i>Intervention A</i>	<i>Intervention B</i>	<i>Outcome happens</i>	<i>Outcome does not happen</i>
On	On	0.9	0.1
On	Off	0.8	0.2
Off	On	0.4	0.6
Off	Off	0.2	0.8

Source: Authors' creation

there is still a 0.1 probability the outcome does not happen, and conversely, when neither intervention is 'on' there is still a 0.2 probability that the outcome does happen. These non-zero probabilities represent 'everything else going on in the system'. They are often an important point of the elicitation process and allow us to capture influence on the outcome, even if we do not formally put them in the network.

You may be wondering why BBNs are Bayesian. They are referred to as 'Bayesian' because of the use of the underlying logic of Bayesian statistics (which provides a way to update probabilities considering new data or evidence) rather than because they were developed by Thomas Bayes himself. Bayesian statistics, simply put, is a field within statistics that revolves around the idea of probability expressing an expectation of likelihood based on prior knowledge or on a personal belief. This probability may be updated based on new information arriving about factors we believe to influence that event. In a sense this operationalises how our belief about a particular probability should change rationally as we learn more. This is in opposition to the Frequentist view of probability which revolves around the idea that probability relates to the relative frequency of an event. We do not want to get into the large and ongoing debates within and between these two schools of thought. However, it is important to recognise that BBNs take that Bayesian idea of probability and implement it through the network structure and conditional probability tables; the parent nodes of any node hold the prior information we are using to update our beliefs about that node. We can also use new information about the states of child nodes to update our beliefs about parent nodes using Bayesian inference.

There is a lot of variety in how BBNs are built, either directly from data, or through participatory processes with experts and stakeholders. However, the object that is produced and the analysis that is done tend to be consistent. There are extensions, such as dynamic BBNs (as mentioned

above), and hybrid BBNs (which allow us to include continuous variables as well as the categorical variables we have described above). Where there is more variety is in the terminology and jargon associated with BBN. Ironically, given the formalism of the method, this is one of the methods with the highest number of different names, but less surprisingly, the opaquest technical language.

You may see BBNs referred to as any of the following: ‘Bayesian networks’, ‘probability networks’, ‘dependency models’, ‘influence diagrams’, ‘directed graphical models’, or ‘causal probabilistic models’. We have also seen them referred to as ‘Theory of Change maps’ because of the similarities with these types of diagrams, that is, a focus on connections between inputs and outcomes, tendency to produce simple maps, and not include feedbacks of many causal influences. This plethora of terms seems to reflect the widespread use of BBNs in different domains rather than large differences in how they are used. Some of the key technical terms you may bump into might include ‘prior probability distribution’, or ‘prior’, which refers to the probability distribution that indicates our best guess about the probability that a state will take some value before new or additional evidence or data is taken into account; and ‘posterior probability distribution’, or ‘posterior’, which is the conditional probability we assign after taking into account new evidence or data. Note that the probability distributions assigned to states of root nodes are prior probabilities because they have no inputs on which their state is conditional.

HOW DO YOU USE BAYESIAN BELIEF NETWORKS?

The main division in constructing BBNs is between approaches which generate the networks and conditional probabilities directly from data, using a range of different learning algorithms, and those which use stakeholder input. Here, we will focus on the latter, though we do consider the issue of developing system maps directly from data in Chap. 9. Even though we are focusing on the participatory mode of BBN, it is worth making clear that you will likely want to use purpose-built BBN software to implement your BBN. You can build the network and collect conditional probability information in standard ways (i.e. workshops, interviews) with general-purpose software or just pen and paper, but you will need the BBN software to run the analysis quickly and easily for you (more on software options at the end of the chapter).

Let’s look at each of the main stages in developing and using BBNs:

- **Build the network:** you have two options at the start, working with stakeholders, you can build either a generic and intuitive causal network (i.e. which may have loosely defined factors, feedbacks, nodes with many parents) or a network which fits closely with the constraints of BBN (i.e. no feedbacks and not too many parents for any one node). Doing the former will make your workshop or interview process marginally easier and more intuitive for stakeholders, but will mean you need to convert the map you start with into a BBN form. This conversion process puts a lot of power and responsibility in the hands of the modeller or researcher, which you may want to avoid. However, it can be a useful iteration step to present back to stakeholders a ‘BBN version’ of a more generic system map they have built, for feedback. For building a network immediately in the form of a BBN, we would recommend you facilitate the process quite strongly, in a similar way to the Theory of Change process in Chap. 3, focusing on the key outcomes and inputs or interventions in the map, and then filling in the gaps with intermediary factors, but making sure you stick within the ‘rules’ of a BBN structure. You may also want to include external controlling factors and factors which are likely to directly influence an intervention’s success. The MERIT guidelines document (Bromley, 2005) gives a useful and detailed guide to BBN construction.
- **Constrain the network:** if you built a more generic system map initially, as is the practice we have observed most often, you will then need to convert it into a BBN form. This will require some big decisions in simplifying the map and constraining it to have no feedbacks or have nodes with many parents (i.e. maximum two or three parents). If you feel the ‘no feedbacks’ issue is too problematic for your system, you may want to consider developing a dynamic BBN with nodes which represent factors at different time points. A simple example of this is shown in Fig. 7.3, where a feedback between ‘wood extraction’ and ‘wood stored’ has been added using wood extraction in two time periods. There is no formal reason to aim for a network of a certain size, but we have tended to see networks with no more than roughly twenty nodes. Given the fact you need to elicit conditional probabilities for each node, it is worthwhile trying to keep the network size small enough that this does not become an overly time-consuming process for your stakeholders.

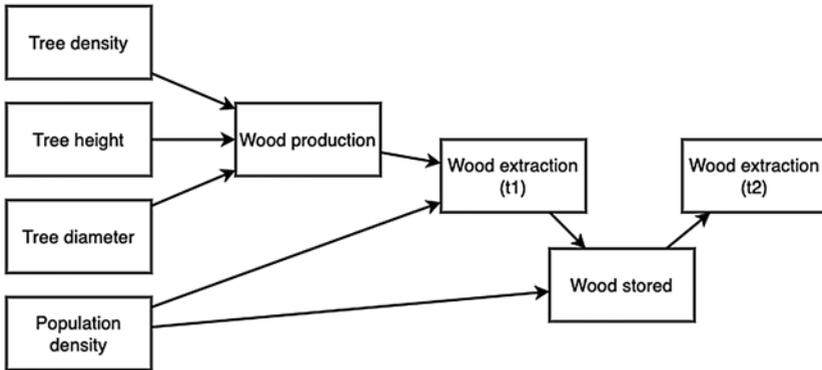


Fig. 7.3 An example dynamic BBN with a feedback between ‘wood extraction’ and ‘wood stored’. (Source: Authors’ creation based on an example in Landuyt et al. (2013))

- Define possible node states:** once you have chosen your nodes, you need to decide what possible states all of them can take. These states need to cover not just what a variable is doing now but how it might change under any conceivable scenario in your model. They must be exhaustive, covering all possibilities, and exclusive, non-overlapping. Although variables in BBNs can be given continuous values, for the most part they are defined as discrete. The options that you have to define node states will depend on the software that you are using. These may take several forms: labels, such as low, medium, or high; Boolean variables; or numbers from a set list or numerical intervals. It is important to find a balance between a number of states that will capture or give the information you need to describe the system and the effect that has on the size of your conditional probability tables.
- Elicit conditional probabilities:** once you have a BBN structure with all the nodes and edges defined, you need to collect conditional probabilities for every node. This is most easily done using conditional probability tables (as in Tables 7.1 and 7.2) for each node. Agreeing on these probabilities, and the number and types of states of nodes, in a workshop setting can be a key point for discussion for stakeholders. You should expect a lot of discussion at this stage, and you may need to update the structure of the map based on this.

- **Analysis:** now that you have your network and conditional probabilities, you can start doing some analysis. You will likely find value in iterating through a few rounds of analysis and discussion with stakeholders when you are working in a participatory mode. In practice, the main ‘site’ for your analysis will be the BBN software. It is normally easy to use the software to manipulate the BBN and address the questions you are interested in; most have simple point-and-click functionality to do this. To start with, your network will default to using the prior probabilities you have specified for the root nodes (i.e. the nodes with no incoming connections) and use these to compute probability distributions for states for the rest of the nodes based on the conditional probabilities you have defined. You can then easily specify the states (or values if you have a hybrid BBN) of any nodes in the network, and the probability distributions of the states of all other nodes will update to reflect this ‘new’ information. By setting states for different combinations of nodes you can create different scenarios and explore their implications in the system. The two most common types of insight we have seen this used for are: (i) to set intervention-type node(s) states and see what this implies for the chances of outcomes being achieved (i.e. what difference does turning an intervention ‘on’ make to the probabilities of an outcome happening?); or (ii) to set outcome-type node(s) states (perhaps based on what we have observed) and see what this implies about the states of other nodes in the network. Once you have explored the BBN and have some analysis, there are a range of ways you can export and present this. It is common to see full networks with states and probabilities defined, as in Fig. 7.2, but others often also choose to show the probabilities of different states for key outcome nodes, given different inputs, perhaps in a simple plot or table.

The stages we have described here are generic. You will need to design a bespoke process that fits your needs. This may emphasise more engagement and iteration with stakeholders or may be more streamlined with a focus on getting to analysis and producing outputs.

COMMON ISSUES AND ‘TRICKS OF THE TRADE’

As with all modelling that produces quantitative outputs, there is a common tendency to over-interpret the precision and reliability of the outputs of BBNs. It is easy for clients or other users of our BBN research to take away a few standalone numbers or plots and focus on these in isolation. Moreover, because BBN deals in probabilities, and is often touted as being useful in contexts of uncertainty (which it is), when results are taken out of context like this, they can still be interpreted as having taken into account uncertainty, creating an undue sense of confidence in using them glibly. We must encourage users to acknowledge that BBNs are always dependent on stakeholder opinion (unless developed based solely on data) and that removing outputs from that context, and not making clear either the process, or the network (i.e. the model), from which they are derived almost always dooms us to see them misinterpreted. Even in cases where outputs are not misused or misunderstood, the appeal of the diagram of a BBN with conditional probabilities annotated can also lead many to view BBN and its associated analysis as a product, rather than a process. Not recognising the value in the process of using this method is to ignore at best half its value, at worst, all its value.

The other common issue relating to the relatively formal nature of BBN is that it can make a participatory process demanding for stakeholders. The process of building a BBN can quickly become a long series of small steps, which individually are dull and/or conceptually dense. This is most common when eliciting conditional probabilities, which can create rather repetitive conversations around filling in conditional probability tables. Even during the building of the underlying network, which is typically one of the more creative and exciting stages in systems mapping, the constraints BBN imposes can become onerous and inhibit discussion, leading to frustration on the part of stakeholders.

To address these, and many other issues, we see four key ‘tricks of the trade’ being used regularly:

- **Emphasise iteration and learning from the process:** do not be afraid to emphasise the need for, and value in, iteration of BBN; build this into stakeholders’, clients’, and users’ expectations. One of the reasons that iteration is so useful is that it allows us to ‘fold-in’ the learning we are generating as we go into the process and the products of BBN.

- **Build a generic map with stakeholders and refine later:** though you can dive straight into building a network which meets the constraints of BBN, we believe that building a more generic causal map, something like a Fuzzy Cognitive Map (Chap. 6) or Participatory System Map (Chap. 5), is likely to be the best way forward. This has the advantages of making that first major stakeholder engagement easier and more flexible, gives you the ability to refine and moderate the network so that it is amenable to the types of analysis you want to do, and creates a useful point for iteration around gathering feedback on the constrained-form network. We believe these advantages outweigh the risk of doing the network refining ‘behind closed doors’, reducing stakeholder engagement, or introducing researcher bias.
- **Tackle criticism of constraints head-on:** in the systems and complexity communities, it is common for people to quickly comment on the constraints BBN introduce. In the most part, these criticisms are not fatal, and often are unfair, so you should develop clear and compelling arguments as to why it is still appropriate to use BBN in your context. Useful retorts include the fact you can have feedbacks if you develop a dynamic BBN, that you can have more parent nodes if you really want to, and that the simplified networks BBN often use hide the nuance and ‘capture-of-context’ that can be done at the conditional probability stage (as we discuss above).

WHAT ARE BAYESIAN BELIEF NETWORKS GOOD AND BAD AT?

Arguably, all the unique strengths of BBN relate to their ability to include probabilistic representations of the states of nodes in causal networks. The mere presence of numbers is not what is useful, rather it is the way the numbers are used. Despite our warnings about abuse of outputs, it is quite difficult to produce meaningless analysis with BBN because it is based on extrapolating individual conditional probabilities (which are easy to define in sensible ways) in a relatively low-risk way, rather than turning the network into a dynamic simulation, which can be fraught with danger. Essentially, unlike a simulation, a BBN retains its transparency even whilst generating numerical output. Everything that has gone into the model is clearly visible in the network structure and probabilities. There are no ‘hidden’ decisions taken by a facilitator or modeller on which the outputs

might depend crucially. (For example, like parameter choices in Fuzzy Cognitive Mapping or System Dynamics.)

The use of probabilities also gives us the wiggle room to bring in wider contexts and causal influences, which it may appear, from the network itself, have been simplified away. Seen in this light, BBNs are one of the better methods for capturing the effect of ‘everything else going on’ in the system in a meaningful way. Because variables are connected by conditional probabilities, there is no need for any mechanistic model of how they change each other’s values. This means that variables of many different types can be brought together in one model, allowing a BBN to cover multiple types of domains important in a problem. Taken together, these strengths hopefully make clear BBN is most useful when we want some quantification, but where it is useful to do this in a probabilistic form, and/or where we want it to be done in a relatively understandable and transparent way.

On the flip side, there are some key weaknesses of BBNs which may be problematic in some settings. They are not good at producing whole-system views of a topic or system and will tend to lead to more simplified models. This is not only in the sense of the breadth and size of map (though size is unconstrained, it is rare to see very large BBNs, i.e. 100 or more nodes) but also in individual connections because of the constraints on the number of parent nodes any one node can have. Where we want this flexibility, or the ability to capture a whole system, BBNs will be less useful. Though dynamic BBNs can capture feedbacks, this is an awkward and potentially incomplete solution (if we only capture one or two cycles of the feedback), so in contexts where there are multiple important feedbacks, BBNs may not be the best choice. Finally, it is worth noting that BBNs can also be time-consuming to update because of the additional conditional probability information that needs to be collected for new nodes, and anything they are connected to. Though they can be updated, in situations where we want a ‘live-document’ or an easily updateable system map, BBNs may be a poor choice.

A BRIEF HISTORY OF BAYESIAN BELIEF NETWORKS

The term ‘Bayesian network’ was first coined by Judea Pearl in his 1988 book, *Probabilistic Reasoning in Intelligent Systems*. Together with Richard Neapolitan’s 1989, and very similarly titled book, *Probabilistic Reasoning in Expert Systems*, the field as it is recognised today was born. These two books reflected much of the thinking that had been developing in the

1980s through a series of workshops (now a conference) on uncertainty in artificial intelligence. Scholars from cognitive science, computer science, decision analysis, medicine, maths and statistics, and philosophy contributed to these developments. As the name of the workshop series suggests, the underlying aim was to incorporate uncertainty into the various types of computational analyses often referred to as ‘knowledge-based systems’ or ‘expert systems’. Put simply, to include uncertainty in these techniques, which were being designed to solve complex problems, or make complex decisions, in human-like ways. These Bayesian networks built on the earlier work of Philip Dawd in the 1970s on the representations of and reasoning with probabilistic dependences. The word ‘belief’ was used in the name of the method right from the start (though not consistently) because this word is used to describe assertions about probabilities in Bayesian statistics. Note, ‘belief’ is not used because BBNs are used in subjective stakeholder participatory processes, though this is a sometimes useful coincidence.

From the start, there were two ways to build BBN, either from data or from stakeholder input. In the early days, data was only used to define conditional probabilities (and then often checked by experts), but not the network structure. As the learning algorithms became more sophisticated, computing power more affordable, and data more plentiful, data could be used to develop the network structure too. More recently, as the value of BBN has been realised in a wider set of fields, and applied to practical policy problems, the use of BBN in a participatory mode has increased in popularity.

GETTING STARTED WITH BAYESIAN BELIEF NETWORKS

We hope you now have a clear picture of what BBN is and how you might use it. The big missing piece we have left undiscussed thus far is software. You will almost certainly want to use some BBN-specific software and there are a few options you have. Let’s now outline some of these software options, and their pros and cons, in Table 7.3. Note, there are more options available than those here; these are just the ones we have used and were recommended to us:

Beyond software, there are resources we would recommend to help you to get an even more detailed understanding of BBN and how to use them, including:

- **Bromley (2005) guidance:** this free and detailed guidance goes into a lot of details on the process of using BBN, including constructing

Table 7.3 BBN software overview

<i>Software</i>	<i>Pros</i>	<i>Cons</i>
AgenaRisk	Commercial package, well supported. Free ‘lite’ version with limited functionality	Cost
BayesiaLab	Commercial package, well supported	Cost
Bayes Server	Commercial package, well supported	Cost
DPL	Commercial package, well supported	Cost Not specifically for BBN but can be used this way
HUGIN Expert	Commercial package, well supported	Cost
GeNIe	Commercial package, well supported Free research and teaching version	Cost (for businesses)
OpenMarkov	Free and open source. Comprehensive functionality	Less stable product, limited support and documentation
Netica—norsys	Commercial package, well supported. Free demo version.	Cost
UnBBayes	Free and open source, simple to use, includes a library of example models	Less functionality than OpenMarkov, limited support

Source: Authors’ creation

them. The example used is water resource management, but it is useful for researchers working in any domain.

- **‘Risk assessment and decision analysis with Bayesian Networks’ (Fenton & Neil, 2018) textbook:** this second edition (first edition from 2013) is practical and focuses on real-world applications of BBN rather than theory.
- **‘Counterfactual and causal inference’ (Morgan & Winship, 2014) textbook:** this book is more focused on the methodology of causal inference, built on the pillars of counterfactuals and causal graphs (which includes BBN). Useful if you want to consider the wider picture, and theory, that BBNs sit within.
- **The Bayesian Network Story (Neapolitan & Jiang, 2016):** this Oxford handbook article provides a detailed overview of the method, which includes a lot of the underlying maths and probability, should you want a clear introduction to that.
- **The ‘classics’:** for those of you who like to return to early texts, it is well worth having a look at the two foundation books for BBN, Pearl (1988) and Neapolitan (1989).

Hopefully, you now have everything you need to get started with BBN. You don't need to have a detailed grasp of Bayesian statistics, but we would recommend you try to get comfortable with the broad ideas. Beyond this, our other priorities before starting would be to play around with one of the software packages above, and to find a couple examples of BBN in similar domains, or used in similar ways, to what you are hoping to do. Good luck!

REFERENCES

- Bromley, J. (2005). Guidelines for the use of Bayesian networks as a participatory tool for Water Resource Management. MERIT. <http://nora.nerc.ac.uk/id/eprint/3300/>
- Fenton, N., & Neil, M. (2018). *Risk assessment and decision analysis with Bayesian networks*. CRC Press.
- Landuyt, D., Broekx, S., D'hondt, R., Engelen, G., Aertsens, J., & Goethals, P. L. M. (2013). A review of Bayesian Belief Networks in ecosystem service modelling. *Environmental Modelling & Software*, 46, 1–11. <https://doi.org/10.1016/j.envsoft.2013.03.011>
- Morgan, S., & Winship, C. (2014). *Counterfactuals and causal inference: Methods and principles for social research*. Cambridge University Press.
- Neapolitan, R. (1989). *Probabilistic reasoning in expert systems*. Wiley.
- Neapolitan, R. & Jiang, X. (2016). *The Bayesian Network Story. The Oxford Handbook of Probability and Philosophy*. <https://doi.org/10.1093/oxfordhb/9780199607617.013.31>.
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

