International Journal of Approximate Reasoning 80 (2017) 475–494

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

A two-phase method for extracting explanatory arguments from Bayesian networks

Sjoerd T. Timmer [a,*], John-Jules Ch. Meyer [a], Henry Prakken [a,b], Silja Renooij [a], Bart Verheij [c]

[a] Utrecht University, Department of Information and Computing Sciences, The Netherlands
[b] University of Groningen, Faculty of Law, The Netherlands
[c] University of Groningen, Artificial Intelligence Institute, The Netherlands

ARTICLE INFO

Article history:
Received 6 November 2015
Received in revised form 5 September 2016
Accepted 6 September 2016
Available online 13 September 2016

Keywords:
Bayesian networks
Argumentation
Probabilistic reasoning
Explanation
Inference
Uncertainty

ABSTRACT

Errors in reasoning about probabilistic evidence can have severe consequences. In the legal domain a number of recent miscarriages of justice emphasises how severe these consequences can be. These cases, in which forensic evidence was misinterpreted, have ignited a scientific debate on how and when probabilistic reasoning can be incorporated in (legal) argumentation. One promising approach is to use Bayesian networks (BNs), which are well-known scientific models for probabilistic reasoning. For non-statistical experts, however, Bayesian networks may be hard to interpret. Especially since the inner workings of Bayesian networks are complicated, they may appear as black box models. Argumentation models, on the contrary, can be used to show how certain results are derived in a way that naturally corresponds to everyday reasoning. In this paper we propose to explain the inner workings of a BN in terms of arguments.

We formalise a two-phase method for extracting probabilistically supported arguments from a Bayesian network. First, from a Bayesian network we construct a *support graph*, and, second, given a set of observations we build arguments from that support graph. Such arguments can facilitate the correct interpretation and explanation of the relation between hypotheses and evidence that is modelled in the Bayesian network.

1. Introduction

Bayesian networks (BNs), which model probability distributions, have proven value in several domains, including medical and legal applications [1,2]. However, the interpretation and explanation of Bayesian networks is a difficult task, especially for domain experts who are not trained in probabilistic reasoning [3]. Legal experts, for example, such as lawyers and judges, may be more accustomed to argumentation-based models of proof because probabilistic reasoning is often considered a difficult task [4,5]. Recently, a scientific interest in combining argumentation-based models of proof with probabilities has arisen [6–10]. One possible combination is the use of argumentation to explain probabilistic reasoning. Argumentation is a well studied topic in the field of artificial intelligence (see chapter 11 of [11] for an overview). Argumentation theory provides models that describe how conclusions can be justified. These models closely follow the reasoning patterns present in human reasoning. This makes argumentation an intuitive and versatile model for common sense reasoning tasks.

* Corresponding author.
E-mail address: s.t.timmer@uu.nl (S.T. Timmer).

http://dx.doi.org/10.1016/j.ijar.2016.09.002
0888-613X/

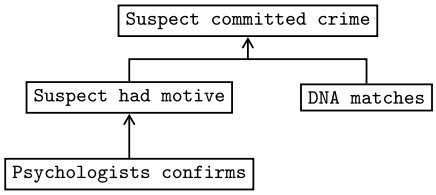

Fig. 1. An example of a complex argument. Every box represents one argument and the arrows show how subarguments support conclusions.

Argumentative explanations of Bayesian reasoning may prove helpful to interpret probabilistic reasoning in legal cases. Existing explanation methods for BNs can broadly be divided in two categories. First, the model itself can be explained. See, for instance, the work of Lacave and Dièz or Koiter [12,13]. Secondly, the evidence can be explained by calculating the so-called *most probable explanation* (MPE) or *maximum a-posteriori probability* (MAP) which is the most likely configuration of a (sub)set of non-evidence variables [14]. A MAP/MPE helps to explain the evidence, but does not explain why the posterior probabilities of variables of interest are high or low nor do they explain the reasoning steps between evidence and hypotheses. In this paper we take a third approach to explaining, which is to explain the derivation of probabilities resulting from the calculations in the BN and explain those using *reasoning chains* that have a clear argumentative interpretation. This resembles the work of Suermondt [15] although that does not apply argumentation, and the work of Schum [16] which is an informal approach to explaining Bayesian networks in argumentative terms. We formalise a method for extracting arguments from a BN, in which we first extract an intermediate support structure, which subsequently guides the argument construction process. This results in numerically backed arguments based on probabilistic information modelled in a BN. We apply our method to a legal example but the approach does not depend on this domain and can also be applied to other fields where BNs are used. Our method thus serves as a general explanation method for BNs.

In earlier work [17] we introduced the notions of probabilistic rules and arguments and a simple algorithm to extract those from a BN. For larger networks, however, this algorithm, which exhaustively enumerates every possible probabilistic rule and argument, is computationally infeasible because it examined inferences between all combinations of variable assignments. We improve on this by searching for explanations in nearby nodes only. Moreover, the algorithm from [17] does unnecessary work because many of the enumerated antecedents will never be met, resulting in irrelevant rules. Similarly, many arguments constructed in this way are superfluous because they argue for irrelevant conclusions from which no further inference is possible. Improving on this work, we proposed a new method that addresses these issues [18]. In this method, the process of argument generation is split into two phases: from the BN, first, a *support graph* is constructed for a variable of interest, from which arguments can be generated in a second phase. This eliminates the aforementioned problem of unnecessarily enumerating irrelevant rules and arguments. As a side-effect this also has the advantage that the support graph is independent of the evidence. When observations are added to the BN, only the resulting argumentation changes. In [18] we introduced an algorithm for the first phase but the second phase was only described informally. In [19] we further formalised the support graph generation phase and we proved a number of properties of this formalism. The current paper further extends [19]. Extensions include the addition of a more elegant and intuitive definition of support graphs and a proof that our algorithm correctly computes such a graph. We have also added a more detailed discussion of the support graph and argument construction method using small examples. Furthermore, we have formalised the second (argument generation) phase and added a case study using an example BN from the literature.

In Section 2 we will present backgrounds on argumentation and BNs. In Section 3 we formally define and discuss support graphs. Using the notion of a support graph we introduce a formalisation of argument construction in Section 4. We apply this method in a case study in Section 5.

2. Preliminaries

2.1. Argumentation

In argumentation theory, one possibility to deal with uncertainty is the use of defeasible inferences. A defeasible rule (as opposed to a strict, or deductive, inference rule) can have exceptions. In a defeasible rule the antecedents do not conclusively imply the consequence but rather create a presumptive belief in it. Using (possibly defeasible) rules, arguments can be constructed. Fig. 1, for instance, shows an argument graph with three nested arguments connected by two rules. From a psychological report it is derived that the suspect had a motive and together with a DNA match this is reason to believe that the suspect committed the alleged crime.

Argumentation can be used to model conflicting or contradictory information. This is modelled by attack between arguments. Undercutting and rebutting attacks between arguments with defeasible rules have been distinguished [20]. A rebuttal attacks the conclusion of an argument, whereas an undercutter directly attacks the inference. An undercutter exploits the fact that a rule is not strict by posing one of the exceptional circumstances under which it does not apply. In this paper we do not use undercutting and *undermining*, which is the third form of attack that can be present in the general case of ASPIC+. The attack relation between arguments can be analysed and from it the acceptability of arguments can be determined.

Different formalisations of argumentation systems exist [21–24]. The formalisation of arguments that we will provide is an instantiation of ASPIC+. We adopt ASPIC+ since it is a state-of-the art formalism for structured argumentation and since it contains all the elements we need, namely, unattackable premisses, defeasible rules and an abstract notion of argument preference which can be instantiated in several ways. By this framework we inherit known results [25] on the rationality postulates that have been developed for structured argumentation [26].

We now describe a simplified version of ASPIC+ because we do not use strict rules, presumed knowledge and we use only one type of attack. For a detailed discussion of this framework we refer the reader to [25]. In ASPIC+ a logical language ($\mathcal{L}$) describes the basic elements that can be argued about. A negation function maps elements of this language to incompatible elements.

Definition 1 *(Argumentation system [25]).* An argumentation system (AS) is a tuple $AS = (\mathcal{L}, \bar{}, \mathcal{R}_d)$ where:

$\mathcal{L}$ is a logical language
$\bar{}: \mathcal{L} \mapsto \mathcal{L}$ is the negation function
$\mathcal{R}_d$ is a set of defeasible inference rules of the form $\varphi_1, \ldots, \varphi_n \Rightarrow \varphi$ (where φ, φ_i are meta-variables ranging over wff in $\mathcal{L}$).

The negation relation over the language can be generalised to something that is called *contrariness*, but we do not require it in this paper.

As described by Pollock [27], defeasible rules are differentiated from strict rules (often denoted $\mathcal{R}_s$ in ASPIC+, although they are not used in this paper) because defeasible rules allow for the existence of exceptions.

To reason with the language and the rules, a knowledge base is required.

Definition 2 *(Knowledge base, after [25]).* In an argumentation system $AS = (\mathcal{L}, \bar{}, \mathcal{R}_d)$, a knowledge base is a set $\mathcal{K}_n \subseteq \mathcal{L}$.

The general case of ASPIC+ distinguishes *axiomatic* knowledge $\mathcal{K}_n$ from *presumed* knowledge $\mathcal{K}_p$ for which $\mathcal{K}_n \cap \mathcal{K}_p = \emptyset$ and $\mathcal{K}_n \cup \mathcal{K}_p = \mathcal{K}$. In such a distinction $\mathcal{K}_n$ cannot be disputed, whereas $\mathcal{K}_p$ can. Since we use the knowledge base to represent the assignments to variables that are observed and we do not wish to dispute observations, we do not use $\mathcal{K}_p$ in this paper. The combination of an argumentation system and a knowledge base forms an argumentation theory.

Definition 3 *(Argumentation theory, [25]).* An *Argumentation theory* is a tuple $AT = (AS, \mathcal{K}_n)$ consisting of an argumentation system AS and a knowledge base $\mathcal{K}_n$.

The contents of the knowledge base $\mathcal{K}_n$ and defeasible rules $\mathcal{R}_d$ is not specified by ASPIC+ and we will define these later for our specific instantiation.

An argumentation theory can be use to build an argument graph by starting with evidence and repeatedly applying rules. ASPIC+ formalises this and defines how these arguments attack each other.

Definition 4 *(Argument, after [25]).* Given an argumentation system AS and a knowledge base $\mathcal{K}_n$, an *argument* A is one of the following:

- ψ if $\psi \in \mathcal{K}_n$, and we define

  $\text{Prem}(A) = \{\psi\}$
  $\text{Conc}(A) = \psi$
  $\text{Sub}(A) = \{\psi\}$
  $\text{TopRule}(A) =$ undefined
  $\text{ImmSub}(A) = \emptyset$
  $\text{DefRules}(A) = \emptyset$

- $A_1, \ldots, A_n \Rightarrow \psi$ if $A_1, \ldots, A_n$ are arguments such that there is a defeasible rule $\text{Conc}(A_1), \ldots, \text{Conc}(A_n) \Rightarrow \psi$ in $\mathcal{R}_d$, and we define

  $\text{Prem}(A) = \text{Prem}(A_1) \cup \ldots \cup \text{Prem}(A_n)$
  $\text{Conc}(A) = \psi$
  $\text{Sub}(A) = \text{Sub}(A_1) \cup \ldots \cup \text{Sub}(A_n) \cup \{A\}$
  $\text{TopRule}(A) = \text{Conc}(A_1), \ldots, \text{Conc}(A_n) \Rightarrow \psi$
  $\text{ImmSub} = \{A_1, \ldots, A_n\}$
  $\text{DefRules}(A) = \text{DefRules}(A_1) \cup \ldots \cup \text{DefRules}(A_n) \cup \{\text{TopRule}(A)\}$

Note that we overload the $\Rightarrow$ symbol to denote an argument while it was originally introduced to denote defeasible inference rules. This is common practice in argumentation and originates from [23]. In such an argument A, ψ is referred to as the conclusion of A which is written as $\text{Conc}(A)$. The last applied rule is referred to as the *top-rule* and written as $\text{TopRule}(A)$. The arguments $A_1, \ldots, A_n$ are called immediate sub-arguments. The notation $\text{ImmSub}(A)$ is used for the set of immediate sub-arguments of A. By sub-arguments $\text{Sub}(A)$ we refer to subarguments of sub-arguments at any depth. By *premises* ($\text{Prem}(A)$) of an argument, we mean all sub-arguments that do not use a rule but an item from the knowledge base. The set $\text{DefRules}(A)$ is used to denote all defeasible rules used in the subarguments of A.

Definition 5 *(Argument terminology, after [25]).* An argument A is said to be *strict* iff $\text{DefRules}(A) = \emptyset$ and *defeasible* otherwise.

An argument can be attacked by *rebutting* it on a conflicting conclusion.

Definition 6 *(Attack).* Argument $\underline{A \text{ attacks}}$ another argument B on $B' \in \text{Sub}(B)$ iff A *rebuts* B on B'. Argument A rebuts argument B on B' iff $\text{Conc}(A) = \overline{\text{Conc}(B')}$.

Informally, an argument rebuts another argument if it is incompatible with one of the intermediate (or final) conclusions. The general case of the ASPIC+ framework specifies two further ways in which arguments can attack each other that we do not use in this paper.

To determine which arguments defeat each other, a binary preference ordering $\preccurlyeq$ is required. We denote the strict version of the ordering as $A \prec B$ when both $A \preccurlyeq B$ and $A \not\succcurlyeq B$ which states that B is strictly preferred over A. We use $A \not\prec B$ to denote that B is not strictly preferred over A. Such an ordering is usually defined on the basis of an ordering of the defeasible rules, but in our case it will be based on a notion of strength that is derived from the probabilities in the BN.

Using an argument ordering, some of the attacks result in defeat of the attacked argument.

Definition 7 *(Argument defeat, after [25]).* Given a collection of arguments $\mathcal{A}$ ordered by an ordering $\preccurlyeq$, a defeat relation $\mathcal{D} \subseteq \mathcal{A} \times \mathcal{A}$ among arguments is defined such that: argument A *defeats* argument B iff A rebuts B on $B' \in \text{Sub}(B)$ and $A \not\prec B'$.

In this way, arguments can be compared on their strengths to see which attacks succeed as defeats. The set of arguments $\mathcal{A}$ and the defeat relation $\mathcal{D}$ can be used as input to Dung's theory of abstract argumentation [28]. On the basis of $(\mathcal{A}, \mathcal{D})$ the acceptability of arguments can be determined. A number of admissible extension semantics have been introduced.

Definition 8 *(Dung extensions, after [28]).* Consider arguments $\mathcal{A}$ and defeat relation $\mathcal{D}$. Any argument $A \in \mathcal{A}$ is *acceptable* with respect to some set of arguments $S \subseteq \mathcal{A}$ iff any argument $B \in \mathcal{A}$ that defeats A is itself defeated by an argument in S. A set of argument S is *conflict free* if none of the arguments in S attack each other. Then, a conflict free set of arguments S:

- is an *admissible* extension iff $A \in S$ implies A is acceptable w.r.t. S;
- is a complete extension iff $A \in S$ whenever A is acceptable w.r.t. S;
- is a preferred extension iff it is a set inclusion maximal complete extension;
- is the grounded extension iff it is the set inclusion minimal complete extension;
- is a stable extension iff it is preferred and every argument outside S is defeated by at least one argument that is in S.

These extensions are all consistent sets of beliefs or points of view that can be taken. The different extensions have different interpretations. The grounded extension, for instance, represents the set of arguments that a rational reasoner should minimally accept.

2.2. Bayesian networks

When dealing with probabilistic evidence, often the likelihood ratio (LR) is used as a measure of probative force. The LR expresses the relation between prior and posterior belief in a hypothesis H being true or false upon observing some evidence $\mathbf{e}$:

$$\frac{P(H = true|\mathbf{e})}{P(H = false|\mathbf{e})} = \frac{P(H = true)}{P(H = false)} \cdot \frac{P(\mathbf{e}|H = true)}{P(\mathbf{e}|H = false)}$$

in which the fraction $P(H = true)/P(H = false)$ is called the prior odds and $P(H = true|\mathbf{e})/P(H = false|\mathbf{e})$ the posterior odds. The ratio $P(\mathbf{e}|H = true)/P(\mathbf{e}|H = false)$ is the LR of this evidence. This formula is often treated as an *update rule* because it can be used to compute probabilities after observing evidence from probabilities before observing that evidence. It is noteworthy that *prior* and *posterior* are notions relative to this evidence. The rule can be invoked multiple times by using the posterior of the first evidence as the prior of the second computation. However, it is required that the second LR

is calculated conditioned on all evidence that is already used in the prior. This makes such an approach ideal if all evidence is independent of each other given the hypothesis, but rather complicated if it is not.

When dealing with large amounts of evidence, often some independence information is available. To exploit known independencies a Bayesian network (BN) can be used [29]. A BN allows for an efficient representation of the independences among evidence, hypotheses and intermediate variables. A BN contains a directed acyclic graph (DAG) in which nodes correspond to stochastic variables. We introduce the following notational conventions for all graphs, including the BN graph. We use $\mathrm{Par}(X)$ for the set of parents of node X and $\mathrm{Cld}(X)$ for the children of X. Descendants and ancestors will be written as $\mathrm{Descendants}(X)$ and $\mathrm{Ancestors}(X)$ respectively. For sets of nodes we will use similar notation in boldface fonts. I.e., $\mathrm{Cld}(\mathbf{X})$ (or $\mathrm{Par}(\mathbf{X})$) denotes the union of the children (or parents) of nodes in a set $\mathbf{X}$. The set $\mathbf{V}$ and variables $V_1, V_2, \ldots$ will exclusively refer to BN nodes whereas $\mathbf{N}$ and $N_1, N_2, \ldots$ will refer to nodes in a support graph as defined in the next section.

Every variable V has a number of mutually exclusive and collectively exhaustive outcomes, denoted by $\mathrm{vals}(V)$. Upon observing the variable, exactly one of the outcomes will become true. Throughout this paper we will consider variables to be binary-valued (boolean in our examples even). The reason for this is that we will use the likelihood ratio to compare arguments. This likelihood ratio can only be used to compare two hypotheses with each other.

Definition 9 *(Bayesian network).* A *Bayesian network* is a pair (G, P) where G is a directed acyclic graph $(\mathbf{V}, \mathbf{E})$, with a finite set of variables $\mathbf{V}$ connected by edges $\mathbf{E} = \{(V_i, V_j) | V_i, V_j \in \mathbf{V} \text{ and } V_i \text{ is a parent of } V_j)\}$, and P is a probability function which specifies for every variable V_i the probability distribution $\mathrm{P}(V_i | \mathrm{Par}(V_i))$ of its outcomes conditioned on the parents $\mathrm{Par}(V_i)$ of V_i in the graph.

A BN models a joint probability distribution with independences among its variables implied by *d-separation* in the DAG [30]. The conditional probability distributions together define a joint probability distributions from which any prior or posterior of interest can be computed. When evidence has been observed, we condition on this evidence $\mathbf{e}$, consisting of a value assignment describing the observed values of the instantiated variables. We say that those variables are *instantiated* to their observed values.

The directions of the arrows have no distinct meaning on their own, but collectively they constrain the conditional independences between variables as captured by d-separation. The concept of d-separation is defined in terms of blocking and chains that can be active or inactive depending on the set of instantiated variables.

Definition 10 *(Chain).* A path in a graph is *simple* iff it contains no vertex more than once. A *chain* in a DAG is a simple path in the underlying undirected graph.

Definition 11 *(Head-to-head node).* A variable V_i is a *head-to-head* node with respect to a particular chain $\ldots, V_{i-1}, V_i, V_{i+1}, \ldots$ in a DAG $G = (\mathbf{V}, \mathbf{E})$ iff both $(V_{i-1}, V_i) \in \mathbf{E}$ and $(V_{i+1}, V_i) \in \mathbf{E}$. I.e., it has two incoming edges on that chain.

Definition 12 *(Blocking chain).* A variable V on a chain c *blocks* c iff either

- it is an uninstantiated head-to-head node without instantiated descendants, or
- it is not a head-to-head node with respect to c and it is instantiated.

A chain is *active* iff none of its variables is blocking it. Otherwise it is said to be *inactive*.

Definition 13 *(d-separation).* Sets of variables $\mathbf{V}_A \subseteq \mathbf{V}$ and $\mathbf{V}_B \subseteq \mathbf{V}$ are *d-separated* by a set of variables $\mathbf{V}_C \subseteq \mathbf{V}$ iff there are no active chains from any variable in $\mathbf{V}_A$ to any variable in $\mathbf{V}_B$ given instantiations for variables $\mathbf{V}_C$.

If, in a given BN model, $\mathbf{V}_A$ and $\mathbf{V}_B$ are d-separated by $\mathbf{V}_C$, then $\mathbf{V}_A$ and $\mathbf{V}_B$ are probabilistically independent given $\mathbf{V}_C$.

An example of a BN is shown in Fig. 2. This example concerns a criminal case with five variables describing how the occurrence of some crime correlates with a psychological report and a DNA matching report. The variables `Motive` and `Twin` model the presence of a criminal motive and the existence of an identical twin. The latter can result in a false positive in a DNA matching test. Since adding further evidence can either create or remove independences, d-separation is a dynamic concept. In the example, instantiating the `Motive` variable will make the psychological report independent of the Crime. On the other hand, observing a DNA match will make the Crime and the presence of a twin dependent, which they were not before.

Head-to-head connections can model intercausal interactions. These interactions occur when two variables can cause the same reaction. In our example `Crime` and `Twin` can both cause the DNA to match. When the DNA match is not observed they are independent of each other, but once the match is observed they become dependent. However, the dependency is a negative correlation, even though the `DNA_match` variable features positive correlations with both parents. This is the case because observing the existence of a twin *explains away* the evidence. In a sense, no further explanation for the DNA match is expected. When the intercausal interaction creates a stronger correlation this is called *explaining in* [31]. In the following we will also require the notions of a Markov blanket and Markov equivalence [32].

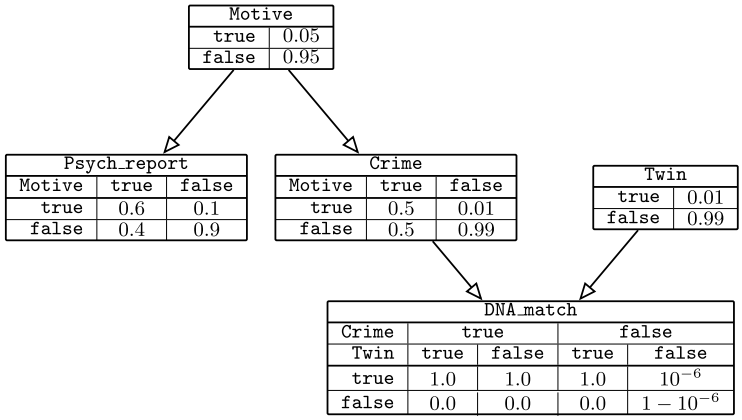

Fig. 2. A small BN concerning a criminal case. The conditional probability distributions are shown as tables inside the nodes of the graph.

Definition 14 *(Markov blanket).* Given a BN graph, the *Markov blanket* $\mathrm{MB}(V_i)$ of a variable V_i is the set

$$\mathrm{Cld}(V_i) \cup \mathrm{Par}(V_i) \cup \mathrm{Par}(\mathrm{Cld}(V_i)) \setminus \{V_i\}$$

I.e., the parents, children and parents of children of V_i (but excluding V_i itself).

The Markov blanket d-separates a node from the rest of the network and is therefore a useful concept.

Definition 15 *(Immorality [29]).* Given a BN graph, an *immorality* is a tuple (V_a, V_c, V_b) of variables such that there are directed edges (V_a, V_c) and (V_b, V_c) in the BN graph but no edges (V_a, V_b) or (V_b, V_a).

Definition 16 *(Graph skeleton).* The *skeleton* of a directed graph is the underlying undirected graph.

Definition 17 *(Markov equivalence [29]).* Two BN graphs are said to be *Markov equivalent* if and only if they have the same skeleton, and the same set of immoralities.

Two Markov equivalent graphs capture the exact same independence relation among their variables. Two BNs with Markov equivalent graphs can therefore describe the exact same joint probability distribution.

3. Support graphs

If a BN is given as input, some evidence can be entered in this probabilistic model and the posteriors can be calculated. However, the results may not be very intuitive to understand. To explain the reasoning from evidence to hypothesis in the Bayesian network, we therefore wish to extract arguments from the BN.

The process of argument generation can be split in two phases. We first construct a support graph from a BN, and subsequently establish arguments from the support graph. In this section we define the support graph and its construction and give an illustration of the construction of a support graph in a small example BN. Moreover, we identify some useful properties of the support graph. The motivation for this graphical transformation from the BN to a support graph is that it abstracts away from the Bayesian network in a way that retains the reasoning chains from the BN. As we will see later, these chains form the skeleton of the arguments, without dealing with evidence yet.

In previous work we developed a method to identify arguments in a BN setting based on exhaustive enumeration of probabilistic rules and rule combinations [17]. A disadvantage of the exhaustive enumeration is the combinatorial explosion of possibilities, even for small models. Using a support graph, we will be able to reduce the number of arguments that needs to be enumerated because only rules relevant to the conclusion of the argument will be considered and we allow rules between variables that are close to each other in the BN. We make this more precise in the next section.

3.1. Definition

Given a BN and a variable of interest $V^\star$, the support graph is a template for generating explanatory arguments. It captures the chains in the BN that end with the variable of interest. As such, it does not depend on observations of variables but rather represents the possible structures in arguments for a particular variable of interest in a particular BN. This means that it can be used to construct an argument based on any set of observations, as we will show in the next section. When new evidence becomes available the support graph can be reused (presuming that the variable of interest does not change).

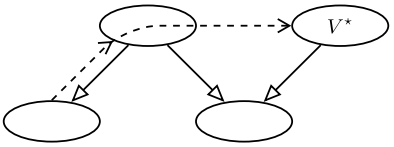

Fig. 3. Illustration of a support chain for variable of interest $V^\star$. The BN edges are solid and have open triangle tips. A possible support chain is shown in dashes and with pointy arrow tips.

This means that the support graph should be able to capture the dynamics in d-separation caused by different observations. Since d-separation is defined on chains we first introduce the notion of a *support chain*.

Definition 18 *(Support chain).* Given a BN $((\mathbf{V}, \mathbf{E}), \mathrm{P})$, a *support chain* for a variable of interest $V^\star \in \mathbf{V}$ is a sequence of variables that:

- follows a simple chain in the BN graph, except that for every immorality (V_i, V_j, V_k) for which V_i, V_j, V_k is on that chain in the BN graph, V_j is skipped in the support chain;
- ends in $V^\star$.

The intuition behind a support chain is that observations of a variable in the BN will propagate through the graph and have some influence on $V^\star$ through the other variables along these chains. From Pearl [33] we know that immoralities can create possible intercausal interactions that deserve special attention: variables that are only connected through a head-to-head connection are a-priori independent. Information should therefore not be propagated through a head-to-head connection. Additional information can, however, create an intercausal dependency. To explicitly capture the possibility of such an induced intercausal relation, we bypass the immoralities in the support chains and create direct links between parents of a common child. In this way every support chain represents a possibly active chain for some set of observations and any chain that is active for some set of observations is represented by a support chain. An example is shown in Fig. 3.

To capture all possible ways in which a variable $V^\star$ can be supported we define the notion of a *support graph*, which can combine multiple support chains. These support chains can be combined in many ways. The definition below defines a family of support graphs that are all valid in the sense that every possible support graph is represented. When used to construct arguments, however, we will see that one specific support graph is exceptionally useful and we provide an algorithm that constructs this (in a sense minimal) support graph.

Definition 19 *(Support graph).* Given a BN $((\mathbf{V}, \mathbf{E}), P)$ and a variable of interest $V^\star \in \mathbf{V}$, a *support graph* is a pair $(\mathcal{G}, \mathcal{V})$ where $\mathcal{G}$ is an acyclic directed graph $(\mathbf{N}, \mathbf{L})$ with nodes $\mathbf{N}$ and edges $\mathbf{L}$, and $\mathcal{V} : \mathbf{N} \mapsto \mathbf{V}$ associates a variable with every node, such that:

$$\mathcal{V}(N_1), \mathcal{V}(N_2), \ldots, \mathcal{V}(N_n) \text{ is a support chain if and only if } N_1, N_2, \ldots, N_n \text{ is a simple, directed path in } \mathcal{G} \text{ with } \mathcal{V}(N_n) = V^\star.$$

We will call N_i a *supporter* of N_j if N_i is a parent of N_j in the support graph, i.e. there is an edge from N_i to N_j.

If the BN graph is multiply connected, a variable may be reachable in more than one way. In that case, it can be associated with more than one of the nodes in the support graph. To distinguish between nodes in the support graph for the same variable, a mapping $\mathcal{V} : \mathbf{N} \mapsto \mathbf{V}$ is introduced that maps support graph nodes to the corresponding variables. When confusion is not possible we will abuse terminology and call V_i a supporter of V_j when we intend to say that N_i is a supporter of N_j for which $\mathcal{V}(N_i) = V_i$ and $\mathcal{V}(N_j) = V_j$.

We will show later how active and inactive chains are treated when we use the support graph to construct arguments about the case. Without knowing which variables are instantiated, the paths in the support graph represent all possibly active chains in the BN.

One of the often misleading aspects of BNs is that directions of individual arrows have no inherent meaning. Sometimes an arrow can be reversed without consequences for the implied independence relation. This is captured by the Markov equivalence property that we mentioned before. One of the advantages of support graphs is that they take away this confusing aspect. Indeed, we can prove that Markov equivalent BNs generate the same support graphs.

Proposition 20. *Given two Markov equivalent BN graphs G and G', and a variable of interest $V^\star$, the sets of support graphs are identical for both BNs.*

Proof. Markov equivalent BN graphs have the same skeleton and the same immoralities. Therefore, they must have the same support chains (which follow the skeleton but bypass immoralities). The set of support chains uniquely defines the possible support graphs, which must therefore be equal. □

Algorithm 1: Recursive algorithm to construct a support graph while building forbidden sets $\mathcal{F}$. Note that, although the order in which the support graph is constructed is not deterministic, the output is not dependent on the order in which nodes are added to the graph because new nodes do not depend on other branches of the already constructed graph.

function `SupportGraphConstruction`$(G, V^\star)$**:**
- **Input**: $G = (\mathbf{V}, \mathbf{E})$ is the BN graph with variables $\mathbf{V}$ and edges $\mathbf{E}$
- **Input**: $V^\star$ is the variable of interest
- **Output**: a support graph $\mathcal{G} = (\mathbf{N}, \mathbf{L})$
- $\mathbf{N} := \{N^\star\}$ $\mathbf{L} := \emptyset$
- $\mathcal{V}(N^\star) := V^\star$
- $\mathcal{F}(N^\star) := \{V^\star\}$
- `expand`$(\mathcal{G}, N^\star)$

function `expand`$(\mathcal{G}, N_i)$**:**
- **Input**: $\mathcal{G} := (\mathbf{N}, \mathbf{L})$ is the support graph under construction
- **Input**: N_i is the support graph node to expand with $V_i = \mathcal{V}(N_i)$
- **foreach** $V_j \in$ `MarkovBlanket`$(\mathcal{V}(N_i))$ **do**
  - **if** $V_j \in$ `Par`$(V_i) \setminus \mathcal{F}(N_i)$ **then** `// case I`
    - $\mathcal{F}_{\text{new}} := \mathcal{F}(N_i) \cup \{V_j\}$
    - `AddSupport`$(\mathcal{G}, N_i, V_j, \mathcal{F}_{new})$
  - **else if** $V_j \in$ `Cld`$(V_i) \setminus \mathcal{F}(N_i)$ **then** `// case II`
    - $\mathcal{F}_{\text{new}} := \mathcal{F}(N_i) \cup \{V_j\} \cup \{V_k | (V_i, V_j, V_k)$ is an immorality$\}$
    - `AddSupport`$(\mathcal{G}, N_i, V_j, \mathcal{F}_{new})$
  - **else if** $V_j \in$ `Par`$(V_k) \setminus \mathcal{F}(N_i)$ *s.t.* $V_k \in$ `Cld`(V_i) **then** `// case III`
    - $\mathcal{F}_{\text{new}} := \mathcal{F}(N_i) \cup \{V_j, V_k\}$
    - `AddSupport`$(\mathcal{G}, N_i, V_j, \mathcal{F}_{new})$

function `AddSupport`$(\mathcal{G}, N_i, V_j, \mathcal{F}_{new})$**:**
- Get from $\mathcal{G}$ a node N_j with:
  - $\mathcal{V}(N_j) = V_j$ and
  - $\mathcal{F}(N_j) = \mathcal{F}_{\text{new}}$
  - or create it if it does not exist in $\mathcal{G}$
- Add (N_j, N_i) to $\mathbf{L}$ in $\mathcal{G}$
- `expand`$(\mathcal{G}, N_j)$

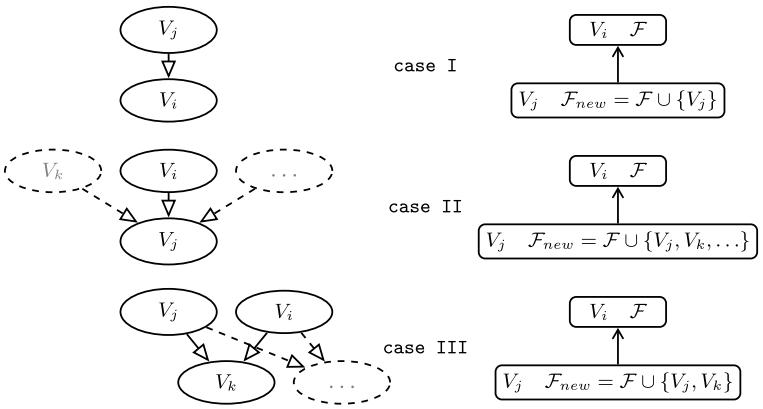

Fig. 4. Visual representation of the three cases in Algorithm 1. A support node for variable V_i can obtain support in three different ways from a variable V_j, depending on its graphical relation to V_i. Note that every support node N_i is labelled with $\mathcal{V}(N_i) = V_i$ and $\mathcal{F}(N_i)$.

A trivial support graph can be constructed by simply enumerating simple chains in the BN and creating a path for every such chain in the support graph, which results in a forest with as many components as there are simple chains in the BN and every such component is a linear path. Since the number of simple chains in a BN is of the order $\mathcal{O}(|\mathbf{V}|!)$ this is not feasible, nor desirable, even for small BNs. Instead, we introduce an algorithm that constructs a more concise support graph in which paths with common prefixes are merged. This algorithm is shown in Algorithm 1 and illustrated in Fig. 4.

The support graph construction algorithm, given in Algorithm 1, uses the notion of a *forbidden set* of variables to maintain a list of variables that should not be used in further support in that branch. This set is used to prohibit the use of, for instance, cyclical reasoning, or reasoning along a head-to-head connection. Fig. 4 shows the three cases of the forbidden set definition. The forbidden set of a new supporter N_i for variable V_i always includes the variable V_i itself, which prevents cyclic traversal of the BN graph and corresponds to the fact that the support graph represents simple chains only.

As we have discussed, in a BN parents of a common child often exhibit intercausal interactions (such as explaining away), which means that if a child has positive correlations with both parents, the parents can be negatively correlated with each other. More generally, the influence between parents may be weaker or stronger, and, in an extreme case, even have the

opposite sign from what we may expect based on the individual influences between the common child and the two parents. Supporting a variable V_i with one of its children V_j and then supporting this child V_j by a parent V_k would incorrectly chain the inferences through a head-to-head node even though an intercausal interaction is possible. Therefore we ensure that this last step cannot be made, by including any other parents that constitute immoralities with a shared child in the second case in the algorithm. A reasoning step that uses the inference according to the intercausal interaction is allowed by the third case of the algorithm. In terms of the support chains this disallows the traversal of a head-to-head connection that is involved in an immorality and it creates the shortcut between the parents of a common child. Note that the use of the intercausal reasoning step requires evidence to be present for the common child. Since the support graph is abstracted from the collection of evidence we allow the step in the support graph, and ensure that the subsequent argument construction verifies that premises and conclusions taken from the support graph are indeed probabilistically dependent.

Theorem 21 *(Correctness). Algorithm 1 creates a support graph* $\mathcal{G} = ((\mathbf{N}, \mathbf{L}), \mathcal{V})$ *for a variable of interest* $V^\star$ *of a BN* $(G = (\mathbf{V}, \mathbf{E}), \mathrm{P})$.

Proof. We prove this in two parts:

1. $\mathcal{V}$ maps every simple directed path in $\mathcal{G}$ ending with the root to a support chain in G, and
2. for every support chain in G, there is a simple path to the root in $\mathcal{G}$ that is mapped to this support chain by $\mathcal{V}$.

Part 1. Any simple directed path in the support graph is constructed from the steps in Algorithm 1 and therefore represents a sequence of nodes in the BN. We need to prove that the sequence of mapped variables is a simple chain in the BN graph where immoralities have been bypassed. By putting the visited variables in the forbidden set $\mathcal{F}$ it is ensured that this sequence is simple. What remains to be shown is that every consecutive pair of support nodes maps to a parent–child pair or a bypass of an immorality, and that no immoralities remain. The former is ensured by the three cases in the algorithm. Every step either goes to a parent or child, creating a parent–child pair in the chain, or to a parent of a child that together form an immorality. The latter (no immoralities remain) follows from the addition of V_k to $\mathcal{F}$ in the second case of the algorithm that makes it impossible to move to a parent after you move to a child in this sequence.

Part 2. A support chain in G is a simple chain in which immoralities have been bypassed. We need to prove that all such chains have a corresponding directed path in the graph found by the algorithm. We prove this by induction. Suppose that, at some point during the construction, the last part of a support chain starting at V_i and ending in $V^\star$ is already represented by a path in the constructed support graph. Then, there is a leaf N_i in the support graph under construction with $\mathcal{V}(N_i) = V_i$. The previous variable on the support chain V_j is not in $\mathcal{F}(N_i)$ because in that case the support chain would either not be simple or contains an immorality. Therefore V_j is added in one of the three cases of the algorithm. Given that the end of every support chain $V^\star$ is added in the first step of the algorithm, this inductively proves that all support chains are found. □

The specific support graph constructed by Algorithm 1 has a number of interesting properties that we will discuss later, but we first present a small step-by-step example of this algorithm to familiarise the reader with the method.

3.2. Example of construction

Let us now consider the example BN from Fig. 2 and take `Crime` as the variable of interest $V^\star$ since, ultimately, that is the variable under legal debate, which models whether or not the crime was committed by the suspect. The construction steps are shown in Fig. 5. We initiate support graph construction by creating one solitary node $N^\star$ with this variable as its root, i.e. $\mathcal{V}(N^\star) = \texttt{Crime}$. The forbidden set for this node is simply {`Crime`} (step 1 in the figure). We then add nodes to the support graph by trying all three extension steps as described above. The `Crime` node has a parent and a child which has another parent, so all three cases apply (exactly once) and we create three supporters in the support graph. First, the `Crime` node can be supported by its parent (`Motive`). This confirms our intuition that the existence of a motive for the suspect affects our belief in the suspect having committed the crime. Secondly, the `Crime` node can also be supported by its child (`DNA_match`) because a match is strong evidence for the suspect's guilt. And thirdly, outcomes of the `Crime` variable may be supported by outcomes of the *parent of a child* node `Twin`. This corresponds to the fact that finding that the suspect has an identical twin explains away the evidence of the DNA match. These three have been added in step 2 of Fig. 5.

Let us now consider the forbidden sets starting with the last supporter (`Twin`). When using a head-to-head connection in the BN to find support, the common child is added to the forbidden set, which then becomes {`Crime`, `DNA_match`, `Twin`}. This eliminates any further support because it covers the entire Markov blanket of the `Twin` node. For the second supporter (`DNA_match`), the forbidden set is exactly the same because now the child is the supporter itself (and is added to $\mathcal{F}$ for that reason) and any other parents (`Twin` in this case) are added to the forbidden set as prescribed by the algorithm. Again, the entire Markov blanket of the `DNA_match` variable is covered by the forbidden set and no further support is possible. For the first supporter that we mentioned (`Motive`), however, one additional supporter can be added. The forbidden set of the support graph node for `Motive` that we created will be {`Crime`, `Motive`}. This means that the child `Psych_report`

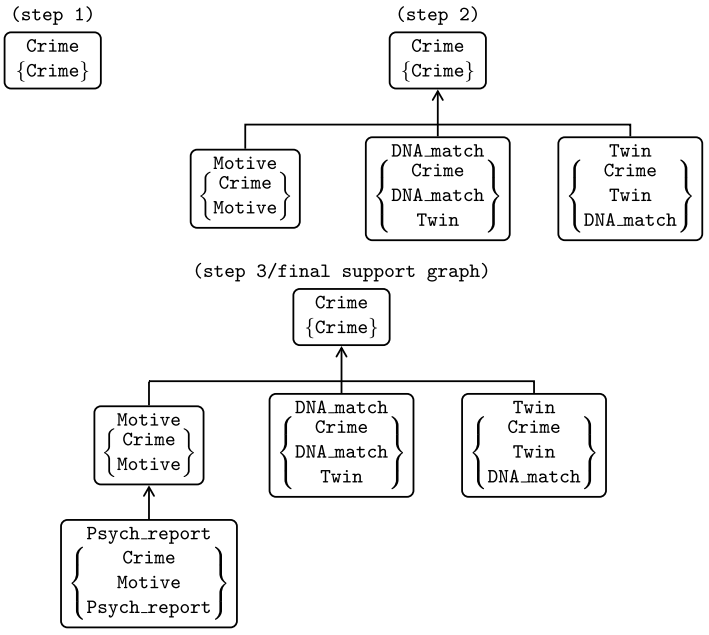

Fig. 5. The steps in the construction of the support graph corresponding to the example in Fig. 2 with $V^\star = \texttt{Crime}$. For every node N_i we have shown the variable name $\mathcal{V}(N_i)$ together with the forbidden set $\mathcal{F}(N_i)$. Multiple edges $(V_i, V), \ldots, (V_k, V)$ into the same node are represented by a hyperedge.

can be used to support outcomes of the `Motive` variable (step 3). This is the result of the fact that the Bayesian network captures the correlation between having a motive and a psychological report on finding this motive. No further support can be added for the `Psych_report` variable and the support graph construction is finished.

3.3. Properties of the support graph algorithm

We now describe some properties of our algorithm to construct support graphs that serve to illustrate the way in which support graphs capture an efficient argumentative representation of what is modelled in a BN.

Property 22. *Given a BN with $G = (\mathbf{V}, \mathbf{E})$, Algorithm 1 constructs a support graph containing at most $|\mathbf{V}| \cdot 2^{|\mathbf{V}|}$ nodes, regardless of the variable of interest.*

Proof. Variables can occur multiple times in the support graph but never with the same $\mathcal{F}$ sets (see the definition). This set contains subsets of other variables and therefore $2^{|\mathbf{V}|}$ is a strict upper bound on the number of times any variable can occur in the support graph. The total number of support nodes is therefore limited by the expression $|\mathbf{V}| \cdot 2^{|\mathbf{V}|}$. □

This is a theoretical upper bound. In practice the number of support nodes will often be significantly smaller when the BN graph is not densely connected. In the special case where the BN is singly connected we can prove that the support graph contains exactly the same number of nodes as the BN.

Definition 23 *(Singly connected graph).* A directed graph is *singly connected* iff the underlying undirected graph is a tree.

Many known graph algorithms that have an exponential worst case running time on multiply connected inputs, have polynomial running times for singly connected graphs. This also holds for our support graph construction algorithm:

Property 24. *Given a BN graph $G = (\mathbf{V}, \mathbf{E})$ and the support graph $\mathcal{G} = (\mathbf{N}, \mathbf{L})$ constructed by Algorithm 1 for some variable of interest. If G is singly connected, every variable occurs exactly once in $\mathcal{G}$ and the size of the support graph is $|\mathbf{N}| = |\mathbf{V}|$.*

Proof. A variable can in theory occur multiple times in the support graph, but this only happens when the graph is loopy (multiply connected). In a singly connected graph there are no loops. This means that using the three available steps from Algorithm 1, the recursive construction encounters every variable exactly once after which it will be forbidden in the ancestors of the resulting support node and unreachable in the BN from any other branch of the support graph. □

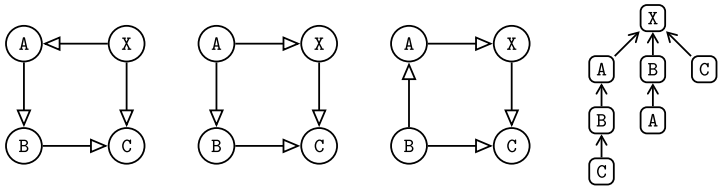

Fig. 6. Three Markov equivalent BNs and their unique support graph for the case that $V^\star = X$.

Specifically, the number of support nodes for a single variable V_i is bounded by the number of simple chains from V_i to $V^\star$ which is smaller for less densely connected graphs. The sparser the BN graph, therefore, the more the support graph will approach size $|V|$.

This shows that the support graph is a concise model to represent the inferences in a BN. We have already seen that support graphs abstract from the sometimes confusing interpretation of the directions of edges. From the bounds on the size of the support graph a bound on the complexity of the algorithm can easily be derived. Specifically, the `expand()` function is called once for every node in the final graph and has itself a worst case complexity of $\mathcal{O}(|\mathbf{V}|)$ because it loops once over the Markov blanket of each variable which could contain all other variables in the graph in the worst case. The worst case complexity of Algorithm 1 is therefore bounded by $\mathcal{O}(|\mathbf{V}|^2 * 2^{|\mathbf{V}|})$ in general and $\mathcal{O}(|\mathbf{V}|^2)$ for singly connected graphs. One of the reasons why BNs are popular as a model for probability distributions is that they provide a considerable reduction in computational power when the graph is not densely connected. A similar improvement holds for our algorithm. In practice, the Markov blankets often contain only a relatively small portion of the other variables in the BN, resulting in fast execution times.

We have already proven in Theorem 25 that two Markov equivalent graphs share the same set of possible support graphs for a specific node of interest. We now show that for our algorithm we can prove that Markov equivalent BN graphs result in a single unique support graph.

Theorem 25. *Given two Markov equivalent BN graphs G and G', and a variable of interest $V^\star$, the two support graphs resulting from Algorithm 1 ($\mathcal{G}$ and $\mathcal{G}'$) are identical.*

Proof. Consider the BN graph G and the corresponding support graph $\mathcal{G}$. In a Markov equivalent graph G' edges may be reversed but not if this creates or removes immoralities. We can prove that the support graphs for G and G' are identical by induction. First the roots have the same variable ($V^\star$) and the same forbidden set ($\{V^\star\}$) by definition. Then, in every iteration of the support graph construction algorithm the added nodes are identical if the support graphs under construction are identical. Following the three possible support steps we see that every supporter follows an edge from the skeleton (which stays the same) or an immorality (which also stays the same). This means that the variables that are associated with the newly added nodes must be the same. If the support graphs were to differ, this has to follow from a different forbidden set. What remains to be shown is that the forbidden sets will also be equal given that the (already found) children have the same forbidden set. Let us consider the three cases of the $\mathcal{F}$ update from Algorithm 1 (see also Fig. 4). Suppose that in the support graph of G, N_i with $\mathcal{V}(N_i) = V_i$ is supporting N_j $\mathcal{V}(N_j) = V_j$:

- if V_j is a parent of V_i in G (case I), then
  - if the direction of the edge in G' is also from V_j to V_i the forbidden sets are trivially the same and
  - if the edge is reversed in G' (from V_i to V_j) then in G' this is handled by case II. This adds any V_k to the forbidden set for which (V_i, V_j, V_k) is an immorality. However, (V_i, V_j, V_k) cannot be an immorality for any V_k in G' because it was not in G and the immoralities are the same.
- if V_j is a child of V_i in G (case II), then
  - if the direction of the edge in G' is also from V_i to V_j the forbidden sets are trivially the same and
  - if the edge is reversed in G' (from V_j to V_i) then in G' this is handled by case I. The forbidden sets are the same except that in G any V_k is added that constitutes an immorality (V_i, V_j, V_k). Again, no such V_k exists because reversal of the edge would not be allowed in G'.
- if (V_i, V_k, V_j) is an immorality (case III), then it must also be an immorality in G' because immoralities in G and G' are the same. Therefore the forbidden sets must also be identical.

Therefore, during the execution of the algorithm on Markov equivalent graphs, the forbidden sets are exactly identical, and therefore the constructed support graphs will be identical. □

What this theorem shows is that Markov equivalent models are mapped to the same support graph, which means that they will receive the same argumentative explanation later on. In Fig. 6, for example, we showed three different but Markov equivalent BNs and the single resulting support graph.

In Section 4 on argument construction the following property is helpful. It states that the support graph constructed by Algorithm 1 is 'minimal' in the sense that support chains have been merged as much as possible. This means that for every support node the set of supporters is 'maximal'.

Theorem 26. *Assume a BN with graph G and a variable of interest $V^\star$. Denote the support graph constructed by Algorithm 1 as $\mathcal{G}$. We have that any two distinct directed paths in $\mathcal{G}$ that end in the root $N^\star$ of $\mathcal{G}$ are mapped to different support chains in G.*

Proof. That such paths map to a support chain was already shown in part 1 of the proof of Theorem 21. That no two simple paths in $\mathcal{G}$ map to the same support chain in Gfollows from the fact that the algorithm never creates multiple support nodes with the same variable and the same forbidden set, and that a support chain uniquely defines the forbidden set (through the 3 cases in the algorithm). Therefore, any support chain in G is represented by one such path in the constructed support graph. □

This is exactly the minimality property of Algorithm 1 that we hinted at earlier. It means that chains in the support graph are merged as much as possible which makes it the most concise support graph among all support graphs that are theoretically possible.

4. Argument construction

From a support graph arguments can be generated that match the reasoning in the BN, since the support graph captures all possible chains of inference. In this section we show how arguments can be generated on the basis of a support graph as constructed by Algorithm 1. We will employ a *strength measure* to rank inferences and to prevent arguments that follow inactive paths in the BN graph.

The interpretation of an argument in this paper is slightly different from what is common in argumentation systems. Since we try to capture the Bayesian network reasoning in arguments, these arguments encapsulate all pro and con reasons for their conclusions. This reflects the way in which Bayesian networks also internally weigh all evidence. The resulting arguments, therefore, do not attack and defeat each other in the way that is common in argumentation. The aim of such an argumentation system is to provide an explanation of the probabilistic reasoning captured by the Bayesian network. The proposed method is not a new form of probabilistic argumentation [34,35], in which probabilities are used to express grades of uncertainty about the arguments. Instead, it is an explanation method for Bayesian network reasoning that translates a Bayesian network to explanatory arguments about the same case. These arguments pose an alternative, qualitative representation of the information represented in and derived from the BN.

Because of our focus on explaining BNs, in our method reasons pro and con a conclusion are combined in a single argument, since in probability theory all evidence has to be considered for drawing conclusions. This is in contrast to the usual modelling of argumentation, in which reasons pro and con a conclusion are distributed over conflicting arguments. Consider, for example, the reasons to believe that a suspect was present at a crime scene at the time of an offence. In other argumentative models, it is usually the case that an argument pro (based on a matching DNA profile that was recovered from the crime scene, for instance) and an argument con (a witness testifying that the suspect was at another location at the time) would result in two arguments. One for the conclusion that the suspect was at the crime scene and one for the conclusion that he/she was not. In our method, however, we find only the argument for one of these conclusions that has both of these premises. Which one we find depends on the probabilities involved. The interpretation of such an argument is that the conclusion holds 'because or despite' the premises. In case of the example above such an argument could be: 'The suspect was at the crime scene because the DNA profiles match, despite the fact that a witness has testified otherwise'.

The arguments that we build will follow the structure of the support graph. As such, the support graph can be seen as a skeleton to build arguments. We present a formal model of these explanatory arguments which instantiates the ASPIC+ framework for structured argumentation. We also discuss how the grounded extension of such a framework can be generated efficiently on the basis of the support graph.

First, we define a logical language $\mathcal{L}$ of sentences used to build arguments. For this language, we take pairs (N, o) of a support node N and one of the outcomes o of the associated variable $\mathcal{V}(N)$. Elements of this language negate each other iff they assign different outcomes to the same variable.

Definition 27 *(Language for explanatory arguments).* Given a BN with graph $G = (\mathbf{V}, \mathbf{E})$ and the corresponding support graph $\mathcal{G} = ((\mathbf{N}, \mathbf{L}), \mathcal{V})$, let the logical language $\mathcal{L}$ be defined as:

$$\mathcal{L} = \{(N, o) \mid N \in \mathbf{N} \text{ and } o \in \text{vals}(\mathcal{V}(N))\}$$

For which the negation is defined as

$$\overline{(N, o)} = (N, o') \text{ such that } o' \in \text{vals}(\mathcal{V}(N)) \text{ and } o' \neq o$$

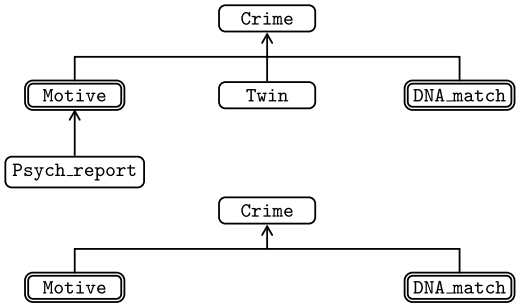

Fig. 7. Support graph from the running example before and after pruning. Instantiated variables are depicted by double node outlines.

Since the support graph captures the allowed paths of reasoning, the rules in the argumentation system should follow the edges of this support graph. When a support node has multiple parents we must consider combinations of supporting parents to form a rule for an outcome of the supported node. In particular, we should consider all parents that can themselves be supported by evidence. This means that we must first consider which chains in the support graph start with actually observed evidence. For this we create a pruned version of the support graph in which all chains start with an instantiated variable and end in the variable of interest.

Definition 28 *(Support graph pruning).* Given a support graph $\mathcal{G}$ for variable of interest $V^\star$ and evidence $\mathbf{e}$ for the BN variables $\mathbf{V}_e$, the *pruned support graph* $\mathcal{G}_\mathbf{e}$ is obtained by repeatedly removing from $\mathcal{G}$ every node N for which either:

- N is an ancestor of a node N' for which $\mathcal{V}(N') \in \mathbf{V}_e$ or
- $\mathcal{V}(N) \notin \mathbf{V}_e$, and N has no unpruned parents.

The second condition resembles the definition of *barren nodes* [29] in a Bayesian network except that nodes are barren iff they are uninstantiated and their *children* are barren.

In Fig. 7 we have depicted the support graph from the running example together with the pruned version for the evidence variables {`Motive`, `DNA_match`}. The node for `Psych_report` has been pruned because it satisfies both conditions (the only path to $V^\star$=`Crime` contains an instantiated variable and it has no unpruned ancestors) and `Twin` has been pruned by the second condition.

The set of defeasible rules is defined to follow the structure of this pruned support graph.

Definition 29 *(Defeasible rules).* Given a support graph $\mathcal{G}$ as constructed by Algorithm 1, observations $\mathbf{e}$ and the pruned support graph $\mathcal{G}_\mathbf{e} = ((\mathbf{N}, \mathbf{L}), \mathcal{V})$, a *rule* in our argumentation system has the form $(N_1, o_1), \ldots, (N_k, o_k) \Rightarrow (N_c, o_c)$ such that

- $N_1, \ldots, N_k$ are all parents of N_c in $\mathcal{G}_\mathbf{e}$, and
- o_c is an outcome of the conclusion variable $\mathcal{V}(N_c)$, and
- $o_1, \ldots, o_k$ are outcomes of the associated variables $\mathcal{V}(N_1), \ldots, \mathcal{V}(N_k)$

These rules are defeasible because they indicate a likely or probable inference rather than a strict deduction.

The evidence that is entered in the BN is represented in the knowledge base.

Definition 30 *(Knowledge base).* Given a Bayesian network and evidence $\mathbf{e}$ for the variables $\mathbf{V}_e$. The knowledge base $\mathcal{K}_n$ contains all observations:

$$\mathcal{K}_n = \left\{ (N_i, o_i) \;\middle|\; \begin{array}{l} \mathcal{V}(N_i) \in \mathbf{V}_e, \text{and} \\ o_i \text{ is logically consistent with } \mathbf{e}, \text{and} \\ (N_i, o_i) \in \mathcal{L} \end{array} \right\}$$

Using Definitions 29 and 30 ASPIC+ specifies arguments, counter arguments and the attack relation. To resolve possible conflicts we consider how arguments can be evaluated against each other. Arguments can attack each other on the outcome of the conclusion variable and defeat can be based on the *strength* of the arguments. To compute this strength any of a number of measures of inferential strength can be used that have been proposed throughout the literature. See for a comparison the work of Crupi [36]. In general, two categories can be identified:

- *incremental* measures of strength assign a number to the weight of the evidence. The Likelihood ratio (LR) is the best known measure of this kind. It expresses the change in the odds of the hypothesis as the result of observing the evidence.

- *absolute* measures assign strength on the basis of posterior probability. The posterior odds measure is a typical example in this class. Such measures capture the a-posterior belief in the hypothesis rather than the change in belief.

In our examples we will use the LR and the posterior odds measures to show how they compare. Note that, although the support graph is not concerned with variable outcomes, the following (and in particular the likelihood ratio as a measure of strength) requires that variables are boolean-valued. Hence we assumed that our input BN contains only binary-valued variables.

Inferential strength can be computed from the BN for every support graph node and depends on the evidence for variables in ancestors of that node in the support graph.

Definition 31 *(Relevant premises to calculate strength).* Consider a support graph $\mathcal{G}_{\mathbf{e}}$ built from a BN with graph $G = (\mathbf{V}, \mathbf{E})$ by Algorithm 1 and pruned to observations $\mathbf{e}$ for the variables $\mathbf{V}_e$.

The set of relevant premises (premises(N_i)) of a support graph node N_i is an assignment to

$$\mathbf{V}_e \cap \left\{ \mathcal{V}(N_j) \middle| N_j \in \text{Ancestors}(N_i) \right\}$$

that is logically consistent with $\mathbf{e}$.

In order to correctly compute the inferential strength, it is important to take into account the correct *context*. This context largely overlaps with the observed evidence. However, instantiations of the variable under consideration are omitted.

Definition 32 *(Context to calculate strength).* Consider support graph $\mathcal{G}_{\mathbf{e}} = ((\mathbf{N}, \mathbf{L}), \mathcal{V})$ built from a BN with graph $G = (\mathbf{V}, \mathbf{E})$ and pruned to observations $\mathbf{e}$ for the BN variables $\mathbf{V}_e$.

The context (context(N_i)) of a support graph node N_i is an assignment to

$$\mathbf{V}_e \setminus \left\{ \left\{ \mathcal{V}(N_j) \middle| N_j \in \text{Ancestors}(N_i) \right\} \cup \{\mathcal{V}(N_i)\} \right\}$$

that is logically consistent with $\mathbf{e}$.

The evidence that overlaps with the ancestors of the node under consideration is excluded during the calculation of the strength because it occludes the potential influence between variables that we wish to detect. I.e., to measure the influence of a DNA match on the guilt hypothesis we must (temporarily) ignore the fact that the DNA match was observed. If we would not do that, the hypothesis would appear to be independent of the DNA match.

Definition 33 *(Likelihood ratio as measure of strength).* Consider a BN with graph $G = (\mathbf{V}, \mathbf{E})$ and a support graph ($\mathcal{G}_{\mathbf{e}} = ((\mathbf{N}, \mathbf{L}), \mathcal{V})$) for the variable of interest $V^\star$ and observations $\mathbf{e}$ for the variables $\mathbf{V}_e$. The LR strength of an assignment $V_i = o$ for a given support graph node N_i with $\mathcal{V}(N_i) = V_i$ is

$$strength_{LR}(V_i, o, N_i) = \frac{P(\text{premises}(N_i) \mid (V_i = o) \wedge \text{context}(N_i))}{P(\text{premises}(N_i) \mid (V_i \neq o) \wedge \text{context}(N_i))}$$

Definition 34 *(Posterior odds as measure of strength).* Consider a BN with graph $G = (\mathbf{V}, \mathbf{E})$ and a support graph ($\mathcal{G}_{\mathbf{e}} = ((\mathbf{N}, \mathbf{L}), \mathcal{V})$) for the variable of interest $V^\star$ and observations $\mathbf{e}$ for the variables $\mathbf{V}_e$. The posterior odds strength of an assignment $V_i = o$ for a given support graph node N_i with $\mathcal{V}(N_i) = V_i$ is

$$strength_{odds}(V_i, o, N_i) = \frac{P(V_i = o \mid \text{premises}(N_i) \wedge \text{context}(N_i))}{P(V_i \neq o \mid \text{premises}(N_i) \wedge \text{context}(N_i))}$$

Strength as defined for assignments to support graph nodes can be lifted to argument strength directly.

Definition 35 *(Argument strength and ordering).* Let A be an argument with $\text{Conc}(A) = (N, o)$. The strength of A is:

$$\text{strength}(A) = \text{strength}(\mathcal{V}(N), o, N)$$

From this an argument ordering follows. $A \preccurlyeq B$ iff either:

- A is strict (premise argument from observation) and B is not, or
- $\text{strength}(A) \leq \text{strength}(B)$

Fig. 8 shows examples of arguments that can be constructed by ASPIC+ from the given definitions of rules and knowledge bases for the running example. Arguments A_1, A_2, A_3 and A_4 together in fact form the grounded extension of this argumentation system. This is because this argument graph uses the maximal set of premises in every inferential step and it assigns the outcomes that are probabilistically best supported. Fig. 8 shows, in addition, a similar argument that uses the

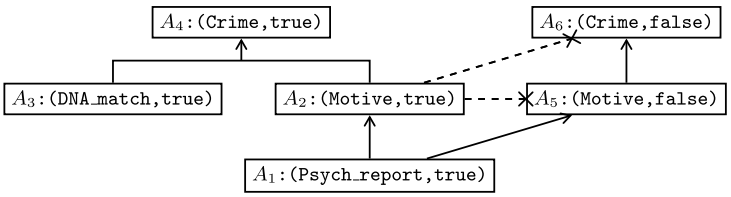

Fig. 8. An argument graph resulting from our running example. Arrows show the immediate sub-argument relation. Besides the intuitively correct arguments $A_1, \ldots, A_4$ there are two additional arguments depicted that can also be made but that are successfully rebutted by A_2. The dashed arrows with crosshair tips show the defeat relation between arguments. Argument A_5 is defeated by A_2 because $(\texttt{Motive}, \texttt{true})$ is probabilistically stronger (using the likelihood ratio measure of strength in this case) than $(\texttt{Motive}, \texttt{false})$ based on this evidence. Any conclusion that builds on this second argument (such as A_6) is also defeated.

same set of premises for that conclusion variable but which draws the 'wrong' conclusion. Such an argument will always be rebutted by the similar argument for the right conclusion. If the two outcomes of the node are equally strong (which in the case of the LR measure of strength means the conclusion is independent of the premises given the evidence), then arguments for both outcomes coexist but defeat each other and will therefore not be part of the grounded extension.In fact, the grounded extension in this argumentation system coincides with the set of undefeated arguments.

Theorem 36. *Consider an argumentation system with the above definitions for the language, rules, knowledge and argument strength. An argument A is in the grounded extension if and only if it is undefeated.*

Proof. Undefeated arguments are by definition part of the grounded extension. For the other way around, we have to prove that any argument in the grounded extension is undefeated. We prove this by induction over subarguments.

For premise arguments, the base case, it is trivially true that they are undefeated because the argument ordering is such that premise arguments are stronger than other arguments and no two premise arguments for different outcomes can exists.

Now for the induction step, we have to prove that an argument A in the grounded extension is undefeated, given the induction hypothesis which states that all immediate subarguments of A are undefeated.

By construction of our argumentation theory, an argument B with the opposite conclusion $\text{Conc}(B) = \overline{\text{Conc}(A)}$ can be constructed which has the same set of proper subarguments as A. Since A is in the grounded extension, there exists a reinstating argument C in the grounded extension that strictly defeats B. By the induction hypothesis we know that C must directly rebut B, since all the subarguments of B are undefeated. This means that $\text{strength}(C) > \text{strength}(B)$. By the definition of argument strength we have that $\text{strength}(C) = \text{strength}(A)$ and consequently $\text{strength}(A) > \text{strength}(B')$ for any B' that directly rebuts A. By the induction hypothesis we know that no subargument of A is defeated and hence, A is undefeated. □

Corollary 37. *For any argument A in the grounded extension with conclusion* $\text{Conc}(A) = (N, o)$ *for variable* $\mathcal{V}(N) = V_i$*, there is no argument B in the grounded extension with* $\text{Conc}(B) = (N, o')$ *such that* $\text{strength}(V_i, N, o) < \text{strength}(V_i, N, o')$*. In other words, if strength is given by the posterior probability, then the arguments in the grounded extension are for those assignments with the highest probability in the BN.*

Because our argumentation theory has no strict rules and no presumed knowledge it follows that any argument ordering is *reasonable* [37]. This means that all known [25] results regarding rationality postulates [26] on ASPIC+ also hold for our argumentation theory.

Important to note is that due to the nature of support graphs there may be paths in the graph that are inactive given the actual evidence and should therefore not be used to reason along. Since d-separation depends on the actual set of evidence and the support graph is meant to capture possible support independent of the actual set of evidence, these irrelevant reasoning paths are still present in the support graph. Only after evaluating the strengths of arguments will these paths explicitly become redundant.

Since the set of rules is directly based on the support graph it is possible to construct the arguments (and in particular the grounded extension) directly, simply by traversing the nodes of the support graph. For every node the 'best' supported argument can be computed using the chosen measure of strength and when both outcomes are equally well supported we immediately know that both outcomes are defeated by the other and not in the grounded extension. This means that the computation of the grounded extension, which is in general computationally hard, can be done efficiently for this argumentation system.

5. Skidding car case study

We will now apply our method to a more realistically sized example. For this, we use the Bayesian network as described by Huygen [38], which is an adaptation from the causal model presented by Prakken and Renooij [39] for a civil legal case

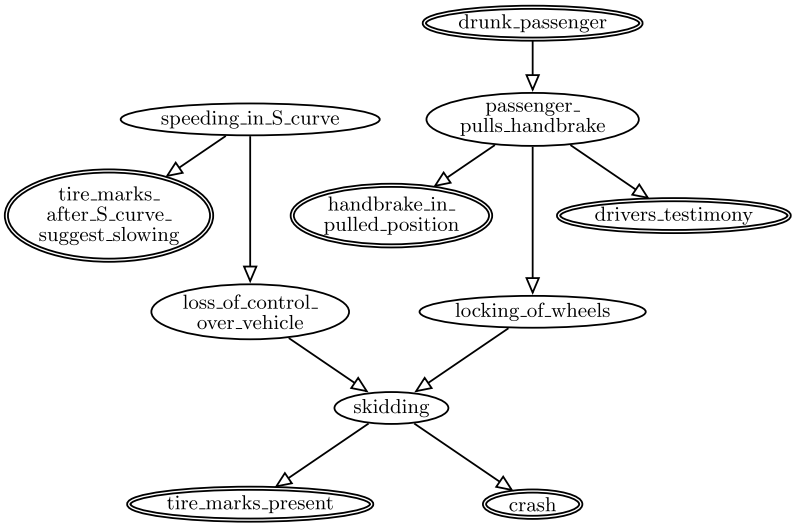

Fig. 9. Graphical structure of the skidding car accident network [38]. Observed outcomes can be distinguished by the double outlines.

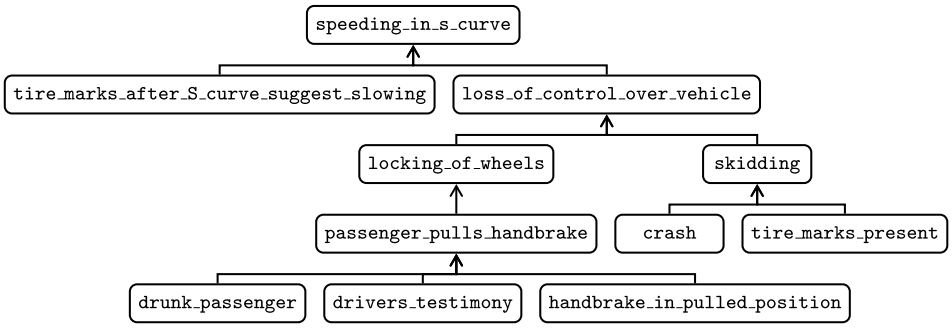

Fig. 10. Support graph from the skidding car accident network.

about a car accident. The graphical structure of this network is shown in Fig. 9. Since the probability tables described by Huygen omit two conditional probabilities we have estimated those in a similar analysis to Huygen's. For our analysis the exact values are not critical. The full specification of the conditional probability tables is given in Appendix A.

5.1. Bayesian network

The example network models the events discussed in an actual legal case about a car accident. The passenger in the car claims that the driver lost control over the vehicle. Because the driver was, supposedly, speeding in the S-curve, the passenger claims that the driver is responsible for the consequences of the accident and wants financial compensation for damages. However, according to the driver it was the passenger (who was drunk at the time of the accident) who pulled the handbrake, causing the car to skid and crash. This case is modelled in eleven variables. Six of these variables are instantiated with evidence. Most importantly, there are tire marks, indicating that the car was skidding before the accident. The nature of the tire marks after the S-curve indicates slowing rather than speeding. Concerning the handbrake, the police found the car with the handbrake in the pulled position. The first thing that the driver said to the police was that the passenger had pulled the handbrake. Finally, it was confirmed by the police that the passenger was drunk at the time of the accident.

5.2. Support graph

Based on this BN, a support graph can be constructed for any of the variables. The variable that we are interested in is `speeding in S curve` because that is what determines the liability of the driver in the accident. The support graph for this variable is shown in Fig. 10.

What can be seen from this support graph is that the observed nature of the tire marks is direct evidence for the fact that the driver was (or was not in this case) speeding in the S-curve. Another supporter for the conclusion is the `loss of control over vehicle` variable because loss of control can occur when one is speeding and has, therefore, a strong correlation with it. The fact that the driver may have lost control over his vehicle is supported by the fact that the car

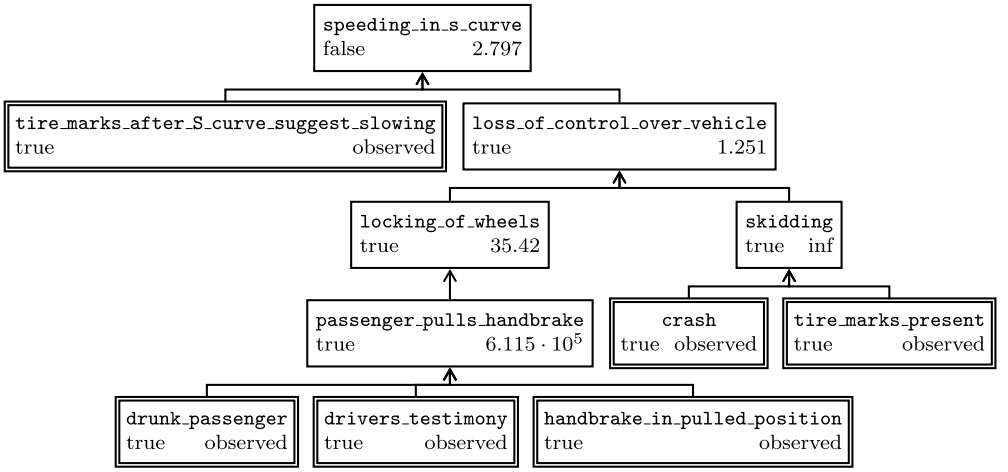

Fig. 11. The best argument for the skidding car accident network using the LR measure of strength. The strengths have been displayed in the nodes that were not instantiated.

was skidding, which in turn is diagnosed by the fact that the crash happened in the first place and the presence of tire marks. The locking of wheels, however, can also explain the skidding and the resulting crash. This may, to some extent, explain away the `loss of control over vehicle` node. The locking of the wheels is supported by the statement that the passenger pulled the handbrake, which is supported by the three observations that the passenger was drunk, that the handbrake was in the pulled position and that the driver testified to the police about this event.

5.3. Arguments

The support graph does not need pruning since all (and only) leaves of the graph correspond to instantiated BN variables. This is because the BN is targeted at this specific set of evidence and no variables have been considered that are irrelevant given the current set of observations.

We first translate the support graph into arguments based on the likelihood ratio measure of inferential strength. The resulting undefeated argument tree is shown in Fig. 11.

We observe that the skidding receives an infinite LR from the evidence below it. This is the case because the probability of finding tire marks was set to 0 if the car did not skid. In other words: there is no other explanation for the tire marks than that the car must have skidded. However, this strong support does not transfer to the `loss of control over vehicle` because the `locking of wheels` (itself with a moderate amount of support) poses an alternative explanation for the skidding. We see that the final conclusion `speeding in S curve = false` is best supported by the combined evidence with a likelihood ratio of 2.797. This does, however, not mean that this conclusion has a high probability, just that it has become more likely by observing the evidence.

To see what the posterior odds are, we can relabel the support graph with posterior odds instead of likelihood ratios. We have done so in Fig. 12. Indeed, we see that the `loss of control over vehicle` has a very low probability, even though the skidding is a certain event (probability 1.0). This shows that both the incremental and absolute reasoning patterns can be explained using this model.

6. Discussion and conclusions

In this paper we formalised a two-phase argument extraction method for explaining the reasoning in BNs. Due to the increase of forensic methods, the legal domain requires that models of evidence are studied that incorporate probabilistic reasoning. One such model is given by Bayesian networks and indeed we find that case specific information has on occasion indeed been modelled as a BN. The question that arises is how these models can be explained. Bayesian networks are notoriously hard to interpret, partly because of the fact that the directions of arrows in a BN have no intrinsic interpretation. We solve this by first extracting a support graph, which removes this confusing property. The support graph is used as the basis for argument construction. This results in qualitative arguments about the same case.

We stress that the proposed method is an explanation tool and cannot be used to replace evidence propagation in Bayesian networks. It rather complements probabilistic inference by representing the same quantitative information in a qualitative, argumentative setting. The resulting argument graphs follow the inference in the BN, but are structured such that they have a clear argumentative interpretation.

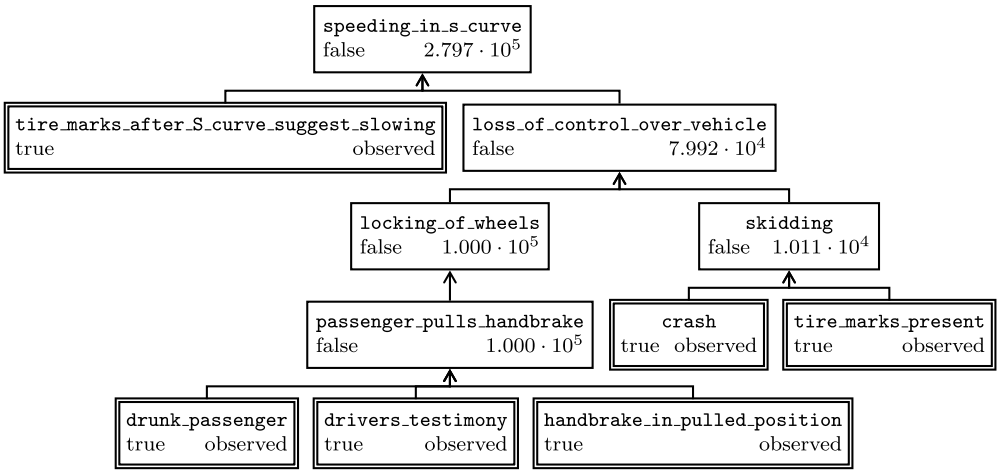

Fig. 12. The best argument for the skidding car accident network labelled with posterior odds of the most likely outcome.

To resolve conflicts between arguments we have applied likelihood ratios and posterior odds. These are always expressed in terms of two alternative hypotheses. It is for this reason that we have limited the approach to binary valued variables. All other aspects of this method do not rely on this. In particular, the general case of ASPIC+ uses *contrariness* which is already a generalisation of negation.

Our support graph method is inspired by the work of Pearl [33], who introduced a logic to deal with causality (and therefore intercausal interactions) in default reasoning. Although that paper is not concerned with Bayesian networks and the extraction of arguments, an important insight that we take from it is that predictive and diagnostic reasoning cannot be chained in a logical approach.

Our approach is similar to Schum's [16], who uses argumentation terminology to understand what is going on in probabilistic analyses of evidential reasoning. However, his notions of argument and argumentation were not formalised, nor related to formal or computational work on argumentation.

Other methods that attempt to explain probabilistic inferences in BNs have focused on visual or textual explanations. See for instance the work of Lacave and Díez [12,40], Koiter [13] and Druzdzel [41]. All of these methods try to make Bayesian networks easier to interpret by visualising correlations or by verbally presenting the relations between variables. However, none of these methods provide (structured) arguments by analysing the reasoning chains in the BN graph using argumentation. In our method we obtain formally well-defined arguments from the Bayesian network. Madigan et al. [42] construct visual explanations of a Bayesian network based on the graphical structure of the BN and a measure similar to the likelihood ratio but they ignore conditional independence captured by the BN and instead first discard the directions of edges. Vreeswijk [43] proposed a simple method to construct rules from BNs to form arguments. His approach, however, only works when there are no head-to-head connections in the BN graph, which is a very limiting constraint on the models that can be used as input.

Another approach to formalise BN reasoning was taken by Keppens [6] who extracts Argumentation Diagrams from BNs. An Argumentation Diagram is a graphical structure that informally represents support between statements, but does not allow one to identify possible counter-arguments. In Argumentation Diagrams, therefore, only one side of the story is highlighted. In a persuasive setting such as legal reasoning, the argumentation approach can be more expressive because it allows for reasoning about the other parties' arguments as well. In our argumentation the pro and con arguments are collectively considered and weighted in accordance with the Bayesian network.

We have shown how support graphs help in the construction of arguments because they capture the argumentative structure that is present in a BN. This provides a general purpose BN explanation method. An advantage of our method is that it offers a dynamic model of evidence: if more observations become available the first phase does not need to be repeated.

Acknowledgements

This work is part of the project "Designing and Understanding Forensic Bayesian Networks with Arguments and Scenarios", supported by the Forensic Science research program financed by the Netherlands Organisation for Scientific Research (NWO). See http://www.ai.rug.nl/~verheij/nwofs/. We also thank the anonymous reviewers for their valuable feedback on this paper.

Appendix A. Conditional probabilities for the skidding car example

drunk_passenger=true	0.05
drunk_passenger=false	0.95

speeding_in_S_curve=true	0.00001
speeding_in_S_curve=false	0.99999

passenger_pulls_handbrake	true	false
locking_of_wheels=true	0.8	0.1
locking_of_wheels=false	0.2	0.9

speeding_in_S_curve	true	false
loss_of_control_over_vehicle=true	1.0	0.00001
loss_of_control_over_vehicle=false	0.0	0.99999

drunk_passenger	true	false
passenger_pulls_handbrake=true	0.03	0.0
passenger_pulls_handbrake=false	0.97	1.0

loss_of_control_over_vehicle	true		false	
locking_of_wheels	true	false	true	false
skidding=true	1.0	1.0	1.0	0.0
skidding=false	0.0	0.0	0.0	1.0

skidding	true	false
crash=true	0.2	0.0001
crash=false	0.8	0.9999

passenger_pulls_handbrake	true	false
drivers_testimony=true	0.9	0.03
drivers_testimony=false	0.1	0.97

passenger_pulls_handbrake	true	false
handbrake_in_pulled_position=true	0.99	0.001
handbrake_in_pulled_position=false	0.01	0.999

speeding_in_S_curve	true	false
tire_marks_after_S_curve_suggest_slowing=true	0.2	0.7
tire_marks_after_S_curve_suggest_slowing=false	0.8	0.3

skidding	true	false
tire_marks_present=true	1.0	0.0
tire_marks_present=false	0.0	1.0

References

[1] F. Taroni, C. Aitken, P. Garbolino, A. Biedermann, Bayesian Networks and Probabilistic Inference in Forensic Science, John Wiley & Sons, Ltd., 2006.
[2] O. Pourret, P. Naïm, B. Marcot (Eds.), Bayesian Networks: A Practical Guide to Applications, Wiley, 2008.
[3] A.P. Dawid, Beware of the DAG!, in: I. Guyon, D. Janzing, B. Schölkopf (Eds.), NIPS Causality: Objectives and Assessment, 2010, pp. 59–86.
[4] D. Kahneman, Thinking, Fast and Slow, Farrar, Straus and Giroux, 2011.
[5] W.C. Thompson, E.L. Schumann, Interpretation of statistical evidence in criminal trials: the prosecutor's fallacy and the defense attorney's fallacy, Law Hum. Behav. 11 (3) (1987) 167–187.
[6] J. Keppens, Argument diagram extraction from evidential Bayesian networks, Artif. Intell. Law 20 (2) (2012) 109–143.
[7] C.S. Vlek, H. Prakken, S. Renooij, B. Verheij, Constructing and understanding Bayesian networks for legal evidence with scenario schemes, in: Proceedings of the 15th International Conference on Artificial Intelligence and Law, ACM, 2015, pp. 128–137.
[8] B. Verheij, To catch a thief with and without numbers: arguments, scenarios and probabilities in evidential reasoning, Law, Probability & Risk 13 (3–4) (2014) 307–325.

[9] N.E. Fenton, D. Berger, D. Lagnado, M. Neil, A. Hsu, When 'neutral' evidence still has probative value (with implications from the Barry George case), Sci. Justice 54 (4) (2014) 274–287.
[10] N.E. Fenton, M. Neil, A. Hsu, Calculating and understanding the value of any type of match evidence when there are potential testing errors, Artif. Intell. Law 22 (1) (2014) 1–28.
[11] F.H. van Eemeren, B. Garssen, E.C.W. Krabbe, A.F.S. Henkemans, B. Verheij, J.H.M. Wagemans, Handbook of Argumentation Theory, Springer, Dordrecht, 2014.
[12] C. Lacave, M. Luque, F.J. Díez, Explanation of Bayesian networks and influence diagrams in Elvira, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 37 (4) (2007) 952–965.
[13] J.R. Koiter, Visualizing inference in Bayesian networks, Master's thesis, Delft University of Technology, 2006.
[14] H.L. Yuan, T. Lu, Most relevant explanation in Bayesian networks, J. Artif. Intell. Res. 42 (2011) 309–352.
[15] H.J. Suermondt, Explanation in Bayesian belief networks, Ph.D. thesis, Departments of Computer Science and Medicine, Stanford University, Stanford, 1992.
[16] J.B. Kadane, D.A. Schum, A Probabilistic Analysis of the Sacco and Vanzetti Evidence, Wiley, New York, 1996.
[17] S.T. Timmer, J.-J.C. Meyer, H. Prakken, S. Renooij, B. Verheij, Extracting legal arguments from forensic Bayesian networks, in: R. Hoekstra (Ed.), Legal Knowledge and Information Systems: The Twenty-Seventh Annual Conference, Vol. 217, JURIX 2014, IOS Press, 2014, pp. 71–80.
[18] S.T. Timmer, J.-J.C. Meyer, H. Prakken, S. Renooij, B. Verheij, A structure-guided approach to capturing Bayesian reasoning about legal evidence in argumentation, in: Proceedings of the 15th International Conference on Artificial Intelligence and Law, ACM, 2015, pp. 109–118.
[19] S.T. Timmer, J.-J.C. Meyer, H. Prakken, S. Renooij, B. Verheij, Explaining Bayesian networks using argumentation, in: S. Destercke, T. Denoeux (Eds.), Proceedings of the 13th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, in: Lecture Notes in Artificial Intelligence, vol. 9161, Springer, 2015, pp. 83–92.
[20] J.L. Pollock, Justification and defeat, Artif. Intell. 67 (1994) 377–407.
[21] S. Modgil, H. Prakken, A general account of argumentation with preferences, Artif. Intell. 195 (2013) 361–397.
[22] G.R. Simari, R.P. Loui, A mathematical treatment of defeasible reasoning and its implementation, Artif. Intell. 53 (1992) 125–157.
[23] G.A.W. Vreeswijk, Abstract argumentation systems, Artif. Intell. 90 (1997) 225–279.
[24] B. Verheij, Deflog: on the logical interpretation of prima facie justified assumptions, J. Log. Comput. 13 (3) (2003) 319–346.
[25] S. Modgil, H. Prakken, The ASPIC+ framework for structured argumentation: a tutorial, Argum. Comput. 5 (1) (2014) 31–62.
[26] M. Caminada, L. Amgoud, On the evaluation of argumentation formalisms, Artif. Intell. 171 (5–6) (2007) 286–310.
[27] J.L. Pollock, Defeasible reasoning, Cogn. Sci. 11 (1987) 481–518.
[28] P.M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artif. Intell. 77 (2005) 321–357.
[29] F.V. Jensen, T.D. Nielsen, Bayesian Networks and Decision Graphs, 2nd edition, Information Science & Statistics, Springer Verlag, 2007.
[30] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Francisco, 1988.
[31] M.P. Wellman, M. Henrion, Explaining 'explaining away', IEEE Trans. Pattern Anal. Mach. Intell. 15 (3) (1993) 287–292.
[32] T. Verma, J. Pearl, Equivalence and synthesis of causal models, in: R. Shachter, T. Levitt, L. Kanal (Eds.), Proceedings of the 6th Conference on Uncertainty in Artificial Intelligence, UAI '90, Elsevier Science Inc., New York, NY, USA, 1991, pp. 255–270.
[33] J. Pearl, Embracing causality in default reasoning, Artif. Intell. 35 (1988) 259–271.
[34] P.M. Dung, P.M. Thang, Towards (probabilistic) argumentation for jury-based dispute resolution, in: Proceedings of the 3rd International Conference on Computational Models of Argument, IOS Press, Amsterdam, 2010, pp. 171–182.
[35] A. Hunter, A probabilistic approach to modelling uncertain logical arguments, Int. J. Approx. Reason. 54 (1) (2013) 47–81.
[36] V. Crupi, K. Tentori, M. Gonzales, On Bayesian measures of evidential support: theoretical and empirical issues, Philos. Sci. 74 (2) (2007) 229–252.
[37] H. Prakken, An abstract framework for argumentation with structured arguments, Argum. Comput. 1 (2010) 1–31.
[38] P.E.M. Huygen, Use of Bayesian belief networks in legal reasoning, in: 17th BILETA Annual Conference, 2002.
[39] H. Prakken, S. Renooij, Reconstructing causal reasoning about evidence: a case study, in: B. Verheij, A.R. Lodder, R.P. Loui, A.J. Muntjewerff (Eds.), Legal Knowledge and Information Systems: The Fourteenth Annual Conference, JURIX 2001, IOS Press, 2001, pp. 131–137.
[40] C. Lacave, F.J. Díez, A review of explanation methods for Bayesian networks, Knowl. Eng. Rev. 17 (2002) 107–127.
[41] M.J. Druzdzel, Qualitative verbal explanations in Bayesian belief networks, AISB Q. 94 (1996) 43–54.
[42] D. Madigan, K. Mosurski, R.G. Almond, Graphical explanation in belief networks, J. Comput. Graph. Stat. 6 (1997) 160–181.
[43] G.A. Vreeswijk, Argumentation in Bayesian belief networks, in: I. Rahwan, P. Moraïtis, C. Reed (Eds.), Argumentation in Multi-Agent Systems, in: Lecture Notes in Computer Science, vol. 3366, Springer, Berlin/Heidelberg, 2005, pp. 111–129.