Argumentation in Bayesian Belief Networks

Gerard A.W. Vreeswijk

Utrecht University, Dept. of Computer Science, The Netherlands
gv@cs.uu.nl

Abstract. This paper establishes an explicit connection between formal argumentation and Bayesian inference by introducing a notion of argument and a notion of defeat among arguments in Bayesian networks.

First, the two approaches are compared and it is argued that argumentation in Bayesian belief networks is a typical multi-agent affair.

Since in theories of formal argumentation the so-called admissibility semantics is an important criterion of argument validity, this paper finally proposes an algorithm to decide efficiently whether a particular node is supported by an admissible argument. The proposed algorithm is then slightly extended to an algorithm that returns the top-k of strongest admissible arguments at each node. This extension is particularly interesting from a Bayesian inference point of view, because it offers a computationally tractable alternative to the $\mathrm{NP}^{\mathrm{PP}}$-complete decision problem k-MPE (finding the top-k most probable explanations in a Bayesian network).

1 Introduction

Bayesian inference and formal argumentation are two important forms of reasoning. Both address the problem of how to reason with uncertain information, and both have developed into major and mature research disciplines. Bayesian inference and argumentation also have strong application areas. Argumentation is slightly biased towards legal applications and Bayesian inference has a tendency towards applications in the medical domain.

Both disciplines share a common goal, but they start from different research hypotheses. The most famous technical difference is that Bayesian inference assumes the availability of a large number of numerical probabilities, while argumentation assumes the opposite, namely, that information on rules and evidence is scarce and qualitative. Besides the technical differences, there is also some sort of cultural gap. On the one hand, proponents of argument systems indicate that realistic problems are often under-specified and ill-formulated. For such problems almost all information is expressed in qualitative terms—provided such information is available at all. Accordingly, proponents of formal argumentation systems argue that argument systems are the best logical means to cope with such problems. On the other hand, proponents of probabilistic reasoning often emphasize that Bayesian inference is the only mathematically correct way to reason with

I. Rahwan et al. (Eds.): ArgMAS 2004, LNAI 3366, pp. 111–129, 2005.
© Springer-Verlag Berlin Heidelberg 2005

uncertain information. Of course both camps are right, it is just that they start from different principles.

Several initiatives have been undertaken to combine Bayesian inference and argumentation [6, 7, 10, 17, 25]. Some of these initiatives use Pearl's probabilistic propagation algorithm as the fundamental notion of support. Other approaches such as [17] propose argumentation features *on top* of Bayesian networks. Finally, there is a recent paper in which an attempt is made to combine Toulmin's argument structures with Bayesian belief networks [1, 20]. However, as far as I know no attempt has been made to import dialectic notions to Bayesian networks, and run true argumentation algorithms on them. This is not done yet, perhaps because argumentation often thwarts probability.

In this paper I try to bridge a part of the gap rather than trying to extend existing formalisms. This means that there is no new theory but rather a proposal to look at Bayesian belief networks from the perspective of argumentation. More specifically, I propose an algorithm that enables users to start an argumentation process within the context of an existing Bayesian belief network. The algorithm possesses a component that is responsible for finding arguments and a component that is responsible for comparing and selecting among the various competing arguments it finds. The corresponding computer program is able to read input files of existing BBN tools (such as Genie) and argue with them in a sensible way.

With the help of the algorithm, I illustrate how one can argue according to conventional argumentation concepts in a Bayesian network, and still be faithful to fundamental probabilistic and dialectic principles.

The algorithm proposed is not a solution to the problem how to translate a defeasible knowledge base into a Bayesian belief network. This is the other direction and will not be discussed here. This paper takes care of the "easy" half of the translation.

The rest of the paper is organized as follows. First some relevant aspects of Bayesian inference and argumentation are reviewed, partially with the help of a simple running example. Then notions of argument and defeat are proposed that have meaning in the context of Bayesian belief networks. Finally, these notions are applied in an algorithm. This algorithm is demonstrated with the help of the earlier example.

2 Bayesian Inference

Bayesian inference is a complex area. This section does not aim to cover this area but discusses only those issues that are relevant here.

Bayesian inference is reasoning within a Bayesian belief network. A Bayesian belief network (BBN) is a finite and directed acyclic graph (DAG) where nodes represent random variables and edges represent probabilistic dependencies among those variables. Most Bayesian belief networks are discrete, in the sense that all random variables can assume only a finite number of states.

2.1 An Example Network

Some of the ideas presented in this paper can best be presented with the help of an example. I have chosen to use a Bayesian network that is made by Gerardina Hernandez in a class homework exercise at the University of Pittsburgh. This network is meant to assess the credit worthiness of an individual (Fig. 2.1). The network is for demonstrational purposes only and is unlikely to be used in real credit assessments. (At least in its present form.)

Fig. 2.1 does not display a general BBN, but a specific case in which there is evidence on Assets, Profession, and Age (bold nodes in the figure). Thus, for this situation we might imagine an applicant of which we only know that his (or her) assets are on the average, that he has a medium-income profession, and that he is aged 28. The dashed node indicates a so-called query node. A query node is simply a node that we are interested in. Here, the query node indicates that we are interested in the credit worthiness of this particular applicant, based on the evidence that we have at hand.

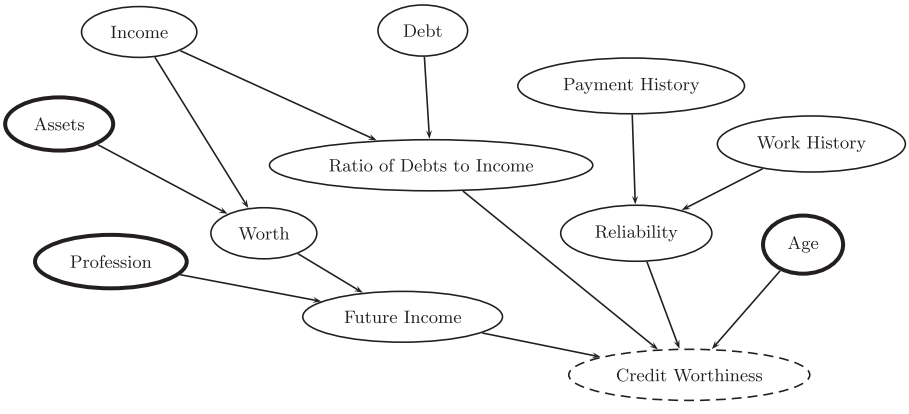

Fig. 1. A sample Bayesian belief network: loan assessment

In other cases, i.e., with other applicants, the evidence may be with other nodes. Sometimes, evidence is complete, in the sense that all prior (i.e., non-conditional) probabilities are overridden by evidence. A simple example of complete evidence is when all leaf nodes are clamped to a particular state. But evidence can also be placed at internal nodes in the network. To exclude trivial cases it is often assumed the set of evidence nodes and the set of query nodes are disjoint.

2.2 Conditional Probability Tables

Probabilistic dependencies among variables is encoded in so-called *conditional probability tables* (CPTs). Each node possesses such a CPT. The CPT of leaf

Table 1. Conditional probability table for future income in decision tree format

High:	Medium:	Low:
High income:	High income:	High income:
Promising: 0.99	Promising: 0.85	Promising: 0.8
Not promising: 0.01	Not promising: 0.15	Not promising: 0.2
Low income:	Low income:	Low income:
Promising: 0.6	Promising: 0.4	Promising: 0.01
Not promising: 0.4	Not promising: 0.6	Not promising: 0.99
Medium income:	Medium income:	Medium income:
Promising: 0.8	Promising: 0.6	Promising: 0.4
Not promising: 0.2	Not promising: 0.4	Not promising: 0.6

nodes are in fact ordinary probability tables (PTs), since such tables encode prior probabilities.

To save space, it is convenient to represent a CPT in a decision tree format (Table 1). In this table, future income can take one of two values: "promising" and "not promising". These two values depend on parent nodes "profession," which takes values "high income," "low income," and "medium income", and "worth" which assumes values "high," "medium," and "low". From the last entry, for instance, we can read that

$$P(\text{ Future income} = \text{Promising } \mid \text{ Worth} = \text{Low } \wedge \text{ Profession} = \text{Medium Income }) = 0.4$$

While CPTs are responsible for representing explicit conditional dependencies, the topology of a BBN itself represents a number of conditional *in*dependencies. One of the consequences of this assumption is that the so-called *joint probabilitiy* of all nodes of a BBN can easily be computed by means of the formula

$$P(x_1, \ldots, x_n) = \Pi_{i=1}^{n} P(x_i \mid \pi(x_i)) \tag{1}$$

where $x_1, \ldots, x_n$ are all nodes of the network in topological order, and where $\pi(x_i)$ are the parents of node x_i.

Although (1) is a well-known and basic result in BBN-theory, the joint probability is generally considered uninteresting, because it chops the probability space into the useless little pieces. For example, to compute $P(x_1)$ in a network with five nodes that each have two states (e.g., true and false), we have to add $2^5/2 = 2^4 = 16$ probabilities, which is a computationally intensive task, at least in the general case. There are algorithms to tackle this task, however, such as the variable elimination algorithm [24].

The reason why joint probabilities are mentioned nevertheless is that they play an important role in the definition of argument strength when arguments are formed in BBNs.

2.3 Evidence Nodes and Query Nodes

In a nutshell, evidence is where reasoning begins and query nodes is where reasoning ends. Evidence is the ultimate stopping place–the things that we think we know, while the query nodes are one or more nodes that we are interested in and from which we want to know their probabilities.

In probabilistic reasoning, a random variable is considered evidence if we know its state. In argumentation, a proposition is considered evidence if it is true. In an argumentation context, evidence is often indicated as a "set of current facts," a "base set" or a "knowledge base".

Thus, each BBN possesses the following node classification.

1. A set of of evidence nodes E. These are nodes that are clamped to a particular state.
2. A set of query nodes X. Often, $|X| = 1$.
3. A set of leaf nodes with a priori probabilities.
4. A set of internal nodes of which the probabilities depend on the node's CPT and the node's parents.

This classification is important because Algorithm 1 below is based on it.

2.4 Bayesian Inference

A recurring task in BBNs is to compute the probability of a collection of query nodes X, given the probabilities that are encoded in the the network by means of CPTs, and given exact values of some observed evidence variables E. The aim of this section is give a brief overview of the complexity of this task.

There are two types of BNN inference tasks: belief updating and belief revision [8]. Many belief updating algorithms can be used for belief revision with just minor modifications, and vice versa.

Belief updating is also called probabilistic inference, or PR. The objective of PR is to compute $P(X|E)$, that is, the posterior probability of query nodes X given evidence E. A simple form of it results when X is a single node. PR typically involves a marginalization operation over query nodes.

The task of belief revision amounts to finding the most probable configuration of some hypothesis variables X, given evidence E. The resulting output is an optimal list of instantiations of X. Almost always $X \cap E = \emptyset$, and in this case the problem is known as computing the Most Probable Explanation, or MPE. A variant of MPE, known as k-MPE, is finding the top-k highest explanations [11]. In the cases when X is a proper subset of all non-evidence nodes, the task is called finding the Maximum a Posteriori Hypothesis, or MAP [3, 11, 18].

Computing PR, MPE and MAP are all NP-hard [3]. However, they still belong to different complexity classes. MPE is a combinatorial optimization problem of which its decision version is NP-complete. PR is harder. It is a counting problem and its complexity is #P-complete [11]. Its decision version is PP-complete. MAP combines both counting and optimization and it is NP^{PP}-complete.

Several algorithms have been proposed to compute the posterior probabilities of query nodes. In the light of the above discussion it should come as no surprise that these algorithms are intrinsically complex. The proposed algorithms have no problems with trees and collections of trees (called polytrees) but if the underlying graph contains multiple paths then some tricks have to be pulled of in order to be able to apply the original algorithm. One of the first Bayesian inference algorithms is Pearl's message passing algorithm [8, 12]. Today, the most commercial tools (such as Genie or Hugin) work with Spiegelhalter junction tree algorithm [4, 9]. For educational purposes the so-called variable elimination method is often used [24].

3 Argumentation

As opposed to Bayesian network theory, the research paradigm of argumentation is less clearly defined. There exist various theories of argumentation and various semantics to interpret a constellation of competing arguments [2, 16].

Nonetheless, a generally accepted view on formal argumentation is Phan Minh Dung's notion of an argument system [5]. Dung's system is an abstract framework that is often used as a stepping stone to define more elaborate systems of argumentation. On the other hand, it is general enough to function as a common divisor of different views on argumentation.

Definition 1 (Argument system, after Dung). *An* argument system *is a directed graph in which the nodes represent arguments, and the arcs between the nodes represent an attack relation among arguments. If a and b are two arguments and $a \leftarrow b$, we say that a is attacked by b.*

Note that the definition says nothing about finiteness of the graph and the absence of cycles or loops (1-cycles).

To design a full-blown argument system, it suffices to specify what an argument looks like and when one argument attacks another argument. Once these two concepts are defined, the argumentation system is defined in its entirety. (We will actually do that in Sec. 5.)

On a more abstract level, Dung's argument system leaves open which arguments we consider valid. (Recall from above that argumentation does not possess a uniform semantics.) One popular and generally accepted notion of argument validity is that of *admissibility*. A set of arguments A is called *admissible* if it satisfies two conditions:

1. *Consistency.* No two arguments in A attack each other.
2. *Self-defence.* Every attacker of an argument in A is attacked itself by an argument in A.

Further, an argument is admissible if it is in at least one admissible set. In this way, an admissible argument might be seen as an argument that belongs to a consistent and complete (read: self-defending) world-view (read: constellation

of admissible arguments).[1] Another view on admissibility is to see it as victory in the competition with other arguments.

Currently, admissibility is the state-of-the-art semantics in theories of formal argumentation.

3.1 An Algorithm to Decide Admissibility

Since the purpose of this work is to present an algorithm to argue in a Bayesian network, I will conclude this section by indicating how a set of admissible arguments can be computed. This information will be used when we define an argumentation algorithm for Bayesian networks.

An algorithm to decide whether an argument a is admissible is relatively simple. It comes down to maintaining a list of arguments L, initially equal to $[a]$, together with an index $1 \leq i \leq \text{length}(L)$, initially set to 0, that indicates up to which index arguments in L are defended by other arguments in L. If a is admissible, then an admissible set can be constructed around a recursively. The algorithm can with one or two modifications in the code (i.e., extremely easily) be cast into a dialectic form, with PRO defending the main thesis and CON trying to hinder PRO's attempt to establish the main thesis [22].

For Dung-type argument systems in which the underlying graph is a-cyclic (most if not all practical systems), the situation is somewhat simpler.

Definition 2 (Defeat). *In a-cyclic and finite argument systems, an argument is* defeated *if it is attacked by an undefeated argument. An argument is* undefeated *if it is not defeated.*

Note that this definition only makes sense in a sub-class of Dung-type argument systems, viz. those argument systems that are a-cyclic and finite.

Since it follows from this definition that arguments without attackers are undefeated, Def. 2 can easily be translated into a simple recursive algorithm. Further, it is a well-known result in the theory of nonmonotonic reasoning and formal argumentation that various different semantics (such as admissibility) reduce to Def. 2 in case the attack graph is a-cyclic. Since BBNs are a-cyclic this is a strong indication for the fact that BBN-type argument systems are a-cyclic (we still have to verify this, of course). Therefore it is reasonable to expect on the basis of the above observations that admissibility in a BBN-type argument systems boils down to Def. 2.

4 Differences

This section lists a number of differences between Bayesian inference and argumentation. Since the two approaches are technically as well as culturally rather

[1] Actually, there exist two versions of admissibility. viz. credulous and skeptical admissibility. An argument is credulously admissible iff it is contained in at least one admissible set. An argument is skeptically admissible iff it is contained in all admissible sets.

different, there are of course a lot of differences to be mentioned. Therefore, this listing is not meant to be exhaustive but instead tries to highlight the differences that are relevant to the algorithm that is going to be proposed.

The most important difference in the light of the forthcoming argumentation algorithm is that BBNs are, what I call, *antecedent-complete* (AC). This means that all evidence for and against every network variable is present in the CPT of the parents of that variable. (Note the words "all" and "every".) Thus, if for example $P(\neg A|\neg B, C)$ is known, then antecedent-completeness guarantees that $P(\neg A|B, C)$, or $P(A|\neg B, C)$ are also known. AC has even more impact if nodes have more than two states (as in the example is the case).

Antecedent-completeness lies at the root of many differences between the two approaches. First, their is the phenomenon of synergy and anti-synergy among parent nodes. There is synergy among two nodes if we can deduce from the knowledge representation that they reinforce each other's support of the child node. For example, two independent witness testimonies typically reinforce the support of the claim that they underpin. Similarly, there is anti-synergy among two nodes if we can deduce from the knowledge representation that they weaken each other's support of the child node. For example, drug A may be a good medication to disease D, drug B may be a good medication to disease D, but their combination may be a less favorable medication to disease D. The fact that BBNs are complete in their antecedents makes them deal correctly with synergy and anti-synergy among parent nodes [12]. This phenomenon is less well mastered in argumentation, where it is called accrual of reasons, or accrual of arguments [15, 21]. There, the question is whether, or under what conditions, reasons should accrue, and if there are general principles behind the accrual of reasons. I maintain that accrual of reasons cannot be modelled in argumentation, because rule bases of argumentation systems are typically antecedent-*in*complete and therefore contain insufficient information as to decide whether there should be synergy or anti-synergy among rules that share consequents.

Another relevant difference between the two approaches is that they have a different view on dealing with new information. A major goal of Bayesian inference is to recompute the probability of all query nodes if evidence is entered into the network. This is a holistic goal, without interest for identifying connections within the network that span multiple nodes. At least Bayesian inference is not interested in explicating such connections. Argumentation, on the other hand, *is* interested in making such connections explicit. The approach of argumentation is to build a case, a "train of reasoning" in support of a claim. This case will do until someone else with other interests builds a case to the contrary.

The latter brings us to another important difference. Within BBNs all evidence is present before an inference is performed. With argumentation, evidence is often produced or retrieved on demand during the argumentation process.

Finally, Bayesian inference often is a one-agent affair. Argumentation, on the other hand, is best performed in a dialogical setting. Arguments are formed not because new evidence comes in (information push) but because parties that

have interest in forming arguments on pain of losing their position in a dispute (information pull).

In conventional argument systems, the question is whether a propositions is defensible, so that a dispute more or less automatically involves two parties, viz. one agent that is trying to defend a claim, and one agent that is trying to punch holes in the defence. In Bayesian networks the situation is somewhat different. There, random variables typically have more than two states so that a scenario is thinkable in which every state (of the same node) is defended by a different agent. Agents do not have to be probabilistically omniscient. CPTs and variables can easily be synchronized along with the queries and query-updates that are sent. In some way this distribution of tasks seems to be remarkably natural, and thus it appears as if argumentation in BBNs is more amenable to multi-agent processing.

5 Argumentation in a Bayesian Belief Network

In this section we will try to interpret elements of BBNs in argumentation-theoretic terms, so as to be able to run an argumentation algorithm on a BBN.

5.1 Arguments

To start with, the CPTs of a BBN contain information that must somehow be reflected in a corresponding argumentation system. Conversely, an argumentation system assumes at the very least a set of rules of inference, because with rules of inference arguments can be formed. A rule of inference can have different forms [13, 14, 16, 19] but with some imagination, the CPTs of the above Bayesian network can be translated into the rule-base and evidence that is displayed in Fig. 2. In this translation, a priori nodes are represented as rules to distinguish them from evidence.

$$
\begin{aligned}
&\text{Rule-base} = \{\\
&\qquad \text{Income(s0-30000)} \leftarrow\!(0.33)\!-\ \text{a-priori}\\
&\qquad \text{Income(s30001-70000)} \leftarrow\!(0.33)\!-\ \text{a-priori}\\
&\qquad \text{Income(s70001-more)} \leftarrow\!(0.33)\!-\ \text{a-priori}\\
&\qquad \text{Debt(a0-11100)} \leftarrow\!(0.33)\!-\ \text{a-priori}\\
&\qquad \ldots\\
&\qquad \text{Ratio(favorable)} \leftarrow\!(0.5)\!-\ \text{Income(s0-30000)} \wedge \text{Debt(a0-11100)}\\
&\qquad \text{Ratio(favorable)} \leftarrow\!(0.001)\!-\ \text{Income(s0-30000)} \wedge \text{Debt(a11101-25900)}\\
&\qquad \ldots\\
&\qquad \text{Ratio(unfavorable)} \leftarrow\!(0.9)\!-\ \text{Income(s30001-70000)} \wedge \text{Debt(a25901-up)}\\
&\qquad \ldots\\
&\qquad \text{Worthiness(positive)} \leftarrow\!(0.9)\!-\ \text{Reliability(reliable)} \wedge \text{Ratio(favorable)} \wedge\\
&\qquad\qquad\qquad\qquad \text{FutureIncome(promising)} \wedge \text{Age(a16-21)}\ \}
\end{aligned}
$$

$$
\text{Evidence} = \{\ \text{Assets(average)}, \text{Profession(Medium-income-profession)}, \text{Age(a16-21)}\}
$$

Fig. 2. Rule-base and evidence corresponding to the loan assessment example

The translation from CPTs to a rule base is conceptually simple and is in fact performed by a small script that rips the CPTs according to the convention that conditional probabilities in XDSL are enumerated such that the rightmost index of the parent state vector runs fastest.

A next step towards argumentation is to chain rules into arguments. In this way, arguments become trees of rules such that roots of trees are query nodes and leaves of trees are evidence or else coincide with the leaves of the network. For example, an argument for the creditworthiness of this particular applicant is

CreditWorthiness(positive) ←(0.80)–
  Reliability(reliable) ←(0.99)–
    PaymentHistory(excellent) ←(0.25)– a-priori
    WorkHistory(stable) ←(0.25)– a-priori
  RatioDebInc(favorable) ←(0.80)–
    Debt(a0-11100) ←(0.33)– a-priori
    Income(s30001-70000) ←(0.33)– a-priori
  FutureIncome(not-promising) ←(0.60)–
    Worth(low) ←(0.70)–
      Income(s0-30000) ←(0.33)– a-priori
      Assets(average)
    Profession(Medium-income-profession)
  Age(a22-65)

Numbers indicate conditional probabilities, taken directly from a BBN's CPTs. Evidence, such as "Assets(average)" is incorporated as unconditional premises of an argument. Similarly, other arguments may be constructed.

The next definition is needed for Def. 4 [attack].

Definition 3 (Sub-argument). *We say that argument a' is a* sub-argument *of argument a if a' is a sub-tree of a such that all leaves of a' are leaves of a.*

One way of looking at sub-arguments is to see them as snapshots of earlier phases in the creation of arguments bottom-up. In particular all premises of an argument a are sub-arguments of a.

5.2 Attack

What remains to be done to obtain a full-fledged argument system, is to define an attack relation between pairs of arguments. To this end, I choose to define the notion of attack on the basis of two notions that are more elementary and (therefore) fall beyond the scope of a Dung-type argument system, viz. the notion of *counterargument* and the notion of *strength* of an argument. First I will discuss counter-arguments, and then I will discuss argument strength.

A Definition of Counter-Argument. In argumentation, the idea is that counter-arguments deny the conclusion of the argument they oppose. In this case, any argument that ends in "¬CreditWorthiness(positive)" would be a counter-argument for the above argument for "CreditWorthiness(positive)". Since nodes

in a BBN are not negated, this is not possible. However, nodes do have different states so we might consider every argument for "CreditWorthiness(x)" with $x \neq$ "positive" as an argument against "CreditWorthiness(positive)".

A Definition of Argument Strength. Sometimes, argumentation is described as "making your case," and this is precisely what happens when one constructs an argument. Thus, an argument might be seen as a description of a concrete case.

To assess the strength of an argument, we will have to look at the likelihood of the entire case supporting the claim in question. This is the joint probability of all argument nodes, except the conclusion. The conclusion is excluded because the degree of belief of the conclusion (the joint probability of the entire argument, including the conclusion) may be close to zero, which indicates a strong argument *against* that conclusion. Such an argument is of informational use only (e.g., can be presented to the user) and will in particular not play a role as a counter-argument in the further defeat among arguments, as such an argument will most likely not be selected as sub-arguments of larger arguments.

In the case that is made in support of "CreditWorthiness(positive)" (cf. the above argument), we end up with a support of 0.00076. This seems to be a disproportional small number. However, the strength of an argument is the probability that the entire case as described by the argument by means of variable instantiation, is realized. This probability is often very small indeed. But since all arguments by definition model specific cases, the competition among arguments remains fair.

A Definition of Attack. On the basis of the notions of counter-argument and argument strength, it is now possible to define a binary attack relation among arguments. The definition that we are going to give is common in the literature of defeasible argumentation [2, 16].

Definition 4 (Attack). *We say that argument a* is attacked by *argument b, written $a \leftarrow b$, if it satisfies the following two conditions:*

1. *Argument b is a counterargument of a sub-argument a' of a.*
2. *Argument b is stronger than argument a'.*

The present notion of attack is defined in terms of counter-argument and argument strength and can thus be implemented in an algorithm.

5.3 Towards an Algorithm

This section discusses the ideas behind the algorithm and utilizes the result that admissibility reduces to defeat in BBN-type argument systems (Sec. 3).

In Sec. 3 it was indicated that there exists a simple algorithm to decide whether an argument is admissible. This suggests that a subroutine to enumerate all arguments for a particular conclusion (preferably in descending order of strength) would suffice to conduct an argumentation process in Bayesian belief

networks. With such a subroutine all arguments and all attackers of all arguments can be found and, once found, conveyed to the algorithm that computes admissible sets. From an argumentation-theoretic point of view, this would be the most logical approach.

The problem with an alternating approach, however, is that such an approach turns out to be extremely wasteful, because BBNs are antecedent-complete (Sec. 4). Antecedent-complete rule-sets make it pointless to search arguments and counter-arguments in separate processes. An approach that better respects the antecedent-completeness of BBNs is to combine the search for arguments and counter-arguments, rather than to conduct search in separate processes. In non-Bayesian argumentation scenario's the latter approach would be intuitive and defensible, but here it is a waste of resources.

The following proposition utilizes the fact that BBNs are a-cyclic and shows that searching for arguments and counter-arguments in parallel still yields admissible arguments. Hence the simpler criterion of Def. 2 can be applied safely.

Proposition 1. *In BBN-type argument systems, an argument a is admissible if and only if*

1. *All immediate sub-arguments of a (i.e., all top-subarguments of a) are undefeated.*
2. *Argument a is the strongest argument for its conclusion node (modulo states) such that (1) holds.*

Proof. First we prove that BBN-type argument systems are are finite and a-cyclic. Finiteness follows from the finiteness of the corresponding BBN. Suppose that there exists an argument system with a cycle $a_1 \leftarrow a_2 \leftarrow \ldots a_n \leftarrow a_1$. If this were the case, then $a_1 \leq a_1' < a_2 \leq a_2' < \ldots < a_n \leq a_{n-1}' < a_1$, where a_i' are the sub-arguments of a_i that are countered by a_{i+1} "$<$" and means "stronger than". This clearly is impossible. The upshot is that we may change the word "admissible" by the word "undefeated," since BBNs are finite and a-cyclic, and from Def. 4 it then follows that argument systems derived from BBNs are finite and a-cyclic.

To prove the proposition, first suppose a is undefeated. We will have to prove that it (1) and (2) hold. To prove the first condition, suppose a' is an immediate sub-argument of an undefeated argument a. [We use a reductio ad absurdum argument because defeat is defined in terms of the existence of defeating arguments.] If a' were defeated, there would be an undefeated argument b against a sub-argument a'' of a'. But in this case b would be a defeater of a as well, which contradicts our earlier assumption. The claim for non-immediate sub-arguments now follows with a simple induction argument. To prove the second condition, suppose that b would be an argument that satisfies (1) but is stronger than a. Then b would fulfill all conditions of being an attacker. Further, since all sub-arguments of b are undefeated, and b is the strongest argument for the conclusion of a (modulo states), b is undefeated. But this would mean that a is defeated by b, which contradicts our earlier assumption.

Conversely, suppose that a is an argument satisfying (1) and (2). We will have to prove that a is undefeated. To this end, let us assume the contrary. Since we may assume that all sub-arguments of a are undefeated by induction, a's defeater must be a counter-argument of a itself. This would imply that this defeater is stronger than a (modulo states). But this would contradict (2). □

The proposition ensures that searching admissible arguments in a BBN amounts to searching for arguments and counter-arguments in parallel, and then simply selecting the strongest argument found.

6 Algorithm

This section explains the algorithm, and finally the algorithm is listed in pseudo-code.

The routine is called recursively at line 4. In order to form the Cartesian product of all sub-arguments, all $A_1, \ldots, A_p$ must be known before the iteration can start at line 6. This implies a considerable memory-overhead. On the other hand, the iteration itself can be executed in an on-demand fashion by means of lazy evaluation. This is what actually has been done in the implementation.

The original algorithm is obtained if $k = 1$ (Alg. 1, line 12). In fact, the algorithm then performs a so-called beam search with beam width $k = 1$. An interesting variation on the original algorithm is obtained if $k > 1$. In that case, the k strongest arguments for each node survive and may act as sub-arguments of possibly larger arguments. The thus obtained variant is an interesting alternative problem statement to the k-MPE problem of finding the top-k most probable explanations in a Bayesian network, because the corresponding decision problem is known to be $\mathrm{NP}^{\mathrm{PP}}$-complete [3, 8].

In this context, it is interesting to determine the time complexity of our argumentation algorithm. We will now do this.

Proposition 2. *The time complexity of Algorithm 1 is* $\mathcal{O}(nsk^p)$.

Proof. First, choose a fixed $k \geq 1$. From this point on we may assume that, for every node, at most k arguments are selected as the top-k of undefeated-arguments-for that node. Suppose a BBN possesses n nodes, and that each node possesses maximally s states and p parents. Let N be a fixed node. At this node the Cartesian product $A_1, \ldots, A_p$ contains at most k^p elements. Since N itself possesses at most s states, at most sk^p arguments need to be considered to determine the top-k of undefeated-arguments-for N. Thus, at worst the strength of at most nsk^p arguments need to considered overall. Since arguments are compared with respect to their strength at every node, and since argument strength is computed once for every argument, the complexity of the entire algorithm is $\mathcal{O}(nsk^p)$. □

Thus, the complexity of Algorithm 1 depends linearly on all graph attributes, except on p. In other words, the complexity of Algorithm 1 is acceptable as long as

Algorithm 1 computing the top-k of undefeated-arguments-for(N)

Require: A node N, a desired state d of N, and an agent X that is interested in strong arguments for $N = d$

if N is clamped to $s_{\text{evidence}} \in \text{states}(N)$ **then**
  $R :=$ the singleton set consisting of
    Argument.new(
      conclusion $\rightarrow N$,
      conclusion-state $\rightarrow s_{\text{evidence}}$,
      degree-of-belief $\rightarrow 1.0$,
      degree-of-support $\rightarrow 1.0$,
      sub-arguments $\rightarrow \emptyset$,
      stakeholder $\rightarrow$ party
)
else
  $A_1, \ldots, A_p = \{$ Undefeated-arguments-for(P) $\mid P$ is a parent of $N\}$
  $R := \emptyset$
  for each argument-vector $(a_1, \ldots, a_p) \in A_1 \times \ldots \times A_p$ **do**
    DOS $:= \Pi_{i=1}^{p}$degree-of-belief(a_i)
    parent-states $:= \{ s_i \mid s_i$ is the state of the conclusion of $a_i\}$
    for each state s of N **do**
      DOB $:=$ DOS $\times$ CPT$_N(s_1, \ldots, s_p, s)$
      $a =$ Argument.new(
          conclusion $\rightarrow N$,
          conclusion-state $\rightarrow s$,
          degree-of-belief $\rightarrow$ DOB,
          degree-of-support $\rightarrow$ DOS,
          sub-arguments $\rightarrow (a_1, \ldots, a_p)$,
          stakeholder $\rightarrow$ party
)
      Extend R with a, removing R's weakest, or one of R's weakest arguments, if $|R| > k$
    end for
  end for
end if
return R

we maintain an upper bound on the number of parents. In practice this is always the case, since the number of entries in a CPT also exponentially depends on p.

Based on the above observations, I think it it safe to claim that the above algorithm is useable for all practical BBNs.

7 Experiments and Results

This section describes how existing case files from the Bayesian network domain are processed. Then, one such file will be run through the argumentation program and the results will be displayed and discussed. Finally, some observations of a more general nature will be made.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<smile version="1.0" id="Credit assesssment" numsamples="1000">
    <nodes>
        <cpt id="PaymentHistory">
            <state id="Excellent" />
            <state id="Aceptable" />
            <state id="NoAceptable" />
            <state id="Without_Reference" />
            <probabilities>0.25 0.25 0.25 0.25</probabilities>
        </cpt>
        <cpt id="Reliability">
            <state id="Reliable" />
            <state id="Unreliable" />
            <parents>PaymentHistory WorkHistory</parents>
            <probabilities>
                0.99 0.01 0.7 0.3 0.7 0.3 0.5 0.5 0.7 0.3 0.55 0.45 0.6
                0.4 0.4 0.6 0.196429 0.803571 0.01 0.99 0.1 0.9 0.01
                0.99 0.7 0.3 0.3 0.7 0.5 0.5 0.2 0.8
            </probabilities>
        </cpt>
   ...
```

Fig. 3. Start of an XDSL input file

The algorithm has been implemented in Ruby 1.8.2, which is an object-oriented scripting language particularly suitable for rapid prototyping. The resulting program has been extended with an XML interface to import Genie 2.0 XDSL files. Genie is a leading software package to create and manipulate decision theoretic models using a graphical user interface. Genie 2.0 stores its networks in a dedicated XML format, named XDSL (Fig. 3).

Besides the network itself (Fig. 3) the argumentation program needs to know which nodes are considered as evidence, and it needs to know the states to which these evidence nodes are clamped. This is specified in the program as an associative array, but we can read this of from Fig. 2. Further, a query node must be specified. This is a node tied to a particular state. In our case the query node is "CreditWorthiness(positive)".

When this program is run on the input as displayed in Fig. 3, we obtain an output as displayed in Fig. 4. From this output, we see that evidence nodes create only one argument, while leaf nodes generate as many arguments as there are states for that node. Since $k = 3$, the algorithm proceeds with at most three strongest arguments at every node and eventually ends up with three arguments in all. The strongest argument supports conclusion "CreditWorthiness(positive)" with sttrength 0.00077 and degree of belief 0.00062. The second strongest argument supports an opposite conclusion with equal support but with less DOB. The second argument is relevant to the main claim but will typically not be used as a sub-argument in further reasoning (if that would happen—here the second argument one of the top arguments).

```
Parsing 'Credit.xdsl' ... done

  1. Searching evidence for CreditWorthiness=Positive
  2. | Searching evidence on both sides for Reliability
  3. | | Searching evidence on both sides for PaymentHistory
  4. | | Search for PaymentHistory=ALL, returns 3 argument(s):
     . . PaymentHistory(excellent) 0.25 1

     . . PaymentHistory(aceptable) 0.25 1

     . . PaymentHistory(noaceptable) 0.25 1

     [snip -- rest of search omitted]

 26. Search for CreditWorthiness=Positive, returns 3 argument(s):
     CreditWorthiness(positive) 0.00062 0.00077
     . Reliability(reliable) 0.062 0.063
     . . PaymentHistory(excellent) 0.25 1
     . . WorkHistory(stable) 0.25 1
     . RatioDebInc(favorable) 0.089 0.11
     . . Debt(a0_11100) 0.33 1
     . . Income(s30001_70000) 0.33 1
     . FutureIncome(not_promissing) 0.14 0.23
     . . Worth(low) 0.23 0.33
     . . . Income(s0_30000) 0.33 1
     . . . Assets(average) 1 1
     . . Profession(medium_income_profession) 1 1
     . Age(a22_65) 1 1

     CreditWorthiness(negative) 0.00015 0.00077
     . Reliability(reliable) 0.062 0.063

     [snip -- rest of argument in output omitted]

     CreditWorthiness(positive) 0.00043 0.00054
     . Reliability(reliable) 0.044 0.063

     [snip -- rest of argument in output omitted]

 27. Ended with 3 arguments.
```

Fig. 4. Output

8 Related Work

Related work falls apart in two categories: probabilistic argumentation systems, such as PAS and hybrid argumentation, and approaches that try to convey dialectic concepts to Bayesian belief networks.

J. Kohlas *et al.* (Fribourg U. Switzerland) work on Probabilistic argumentation systems (PASs). According to their creators, PASs are a form of assumption-

based reasoning for obtaining arguments that support hypotheses [7, 6]. PASs are obtained from propositional logic by considering two disjoint sets $P = \{p_1, \ldots, p_n\}$ and $A = \{a_1, \ldots, a_m\}$ of propositions. The elements of A are called assumptions. $L_{A \cup P}$ denotes the corresponding propositional language. If η is a propositional sentence in $L_{A \cup P}$, then a triple $\mathcal{AS} = (\eta, P, A)$ is called propositional argumentation system and η is called the knowledge base of $\mathcal{AS}$. The knowledge base η is often given as a conjunctive set $\Sigma = \varphi_1, \ldots, \varphi_r$ of clauses $\varphi_i \in \mathcal{D}_{A \cup P}$, where $\mathcal{D}_{A \cup P}$ represents the set of all possible clauses over $A \cup P$. The authors claim that PASs are applicable public-key cryptography and so-called webs of trust, which are essentially holistic and cyclic. Unlike BBNs, the Kohlas *et al.* claim that PASs are able to deal with such cyclic graphs, which is for example essential in the domain of public-key cryptography.

Between 1997-2000, work has been reported on the NAG ("The Nice Argument Generator"), Monash U. Australia [25, 10]. The NAG is an architecture to enable the generation of natural language arguments from BBNs, so that users can argue with the NAG about the implications of various BBN scenario's. Because the NAG is not only concerned with logics but also with user interaction, it consists of several components, such as an argument generator, a strategist, an analyzer, a presenter, an attentional mechanism and an interface. Since the formation of arguments takes place within the NAG's analyzer, only this part of the NAG seems to be relevant within the context of the present paper. Work in which the NAG is reported is not very detailed about the argument formation algorithm. It is mentioned that "the Analyzer performs BN propagation on the portions of the normative and user models which correspond to the Argument Graph and are connected to the goal". Further on in [25], it is mentioned that the NAG applies a Pearl-type propagation algorithm [12]. In [10] a "Generation-Analysis Algorithm" is given in rather detailed terms, but not detailed enough to see how it exploits Pearl's propagation algorithm.

The NAG is partially influenced by Vreeswijk's interactive argumentation system IACAS [23]. Like IACAS, the NAG allows the user to manipulate underlying scenarios. In addition, the NAG is also able to model attentional focus and tailor its arguments to the user in the course of a dialogue.

Finally, there is recent work on modelling argumentation with belief networks, in which an attempt is made to convey Toulmin's argument structures (claim, datum, reason, warrant, backing) to BBNs [1, 20]. This work still is in its preliminary stages but the initial results look promising and demand further research.

9 Conclusion

We have introduced an algorithm with which it is possible to conduct an argumentation process within existing Bayesian belief networks.

The extended algorithm with bean size k is particularly interesting from a Bayesian inference point of view, because it offers a computationally tractable alternative to the $\mathrm{NP}^{\mathrm{PP}}$-complete decision problem k-MPE.

Acknowledgement. This research was supported in part by a European Commission STReP grant ASPIC IST-FP6-002307. This project aims to develop re-usable software components for argumentation-based interactions between autonomous agents.

References

1. V. Carofiglio. Modelling argumentation with belief networks. In F. Grasso, Chris Reed, and Giuseppe Carenini, editors, *Proc. of the 4th Workshop on Computational Models of Natural Argument (CMNA-2004)*, 2004.
2. C.I. Chesñevar, A.G. Maguitman, and R.P. Loui. Logical models of argument. *ACM Comput. Surv.*, 32(4):337–383, 2000.
3. G.F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.
4. R.G. Cowell, A.P. Dawid, S.L. Lauritzen, and D.J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer-Verlag, 1999.
5. Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.
6. Jürgen Kohlas. Probabilistic argumentation systems a new way to combine logic with probability. *Journal of Applied Logic*, 1(3-4):225–253, 2003.
7. Jürgen Kohlas and Rolf Haenni. Assumption-based reasoning and probabilistic argumentation systems. In Jürgen Kohlas and S. Moral, editors, *Defeasible Reasoning and Uncertainty Management Systems: Algorithms*. Oxford University Press, 1996.
8. Kevin B. Korb and Ann E. Nicholson. *Bayesian Artificial Intelligence*. Chapman and Hall/CRC Computer Science and Data Analysis, 2003.
9. S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their applications to expert systems. *Journal of the Royal Statistical Society, B*, 50:157–224, 1988.
10. Richard McConachy, Kevin B. Korb, and Ingrid Zuckerman. A Bayesian approach to automating argumentation. In David M. W. Powers, editor, *Proc. of the Joint Conf. on New Methods in Language Processing and Computational Natural Language Learning: NeMLaP3/CoNLL98*, pages 91–100. Association for Computational Linguistics, Somerset, New Jersey, 1998.
11. J. D. Park. Map complexity results and approximation methods. *Proc. of the 18th Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 388–396, 2002.
12. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, Inc., Palo Alto CA, 2 edition, 1994.
13. John L. Pollock. *Cognitive Carpentry. A Blueprint for How to Build a Person*. MIT Press, Cambridge, MA, 1995.
14. John L. Pollock. Implementing defeasible reasoning. Presented at the Computational Dialectics Workshop, at FAPR'96, June 3-7, 1996, Bonn. Cf. http://nathan.gmd.de/ projects/zeno/fapr/programme.html., 1996.
15. John L. Pollock. Defeasible reasoning with variable degrees of justification. *Artificial Intelligence Journal*, 133(1-2):233–282, 2001.
16. H. Prakken and Gerard A.W. Vreeswijk. Logics for defeasible argumentation. In D.M. Gabbay et al., editors, *Handbook of Philosophical Logic*, pages 219–318. Kluwer Academic Publishers, Dordrecht, 2002.

17. S. Saha and S. Sen. A bayes net approach to argumentation based negotiation. In *Proc. of the AAMAS-2004 workshop on Argumentation in Multi-Agent Systems (ArgMAS-2004)*, 2004.
18. S. E. Shimony. Finding maps for belief networks is np-hard. *Artificial Intelligence*, 68:399–410, 1994.
19. G.R. Simari and R.P. Loui. A mathematical treatment of defeasible argumentation and its implementation. *Artificial Intelligence*, 53:125–157, 1992.
20. Stephen Toulmin. *The Uses of Argument*. Cambridge University Press, 1985.
21. H. Bart Verheij. Accrual of arguments in defeasible argumentation. In C. Witteveen, W. van der Hoek, J.-J. Ch. Meyer, and B. van Linder, editors, *Proc. of the 2nd Dutch/German Workshop on Nonmonotonic Reasoning*, pages 217–224, Utrecht, 1995.
22. G. Vreeswijk and Prakken H. Credulous and sceptical argument games for preferred semantics. In M. Ojeda-Aciego, I.P. de Guzman, G. Brewka, and L.M. Pereira, editors, *Proceedings of the 7th European Workshop on Logics in Artificial Inteligence (JELIA 2000)*, volume 1919 of *Lecture Notes in Artificial Intelligence*, pages 239–253. Springer-Verlag, Heidelberg, 2000.
23. Gerard A.W. Vreeswijk. IACAS: An implementation of Chisholm's principles of knowledge. In C. Witteveen, W. van der Hoek, J.-J. Ch. Meyer, and B. van Linder, editors, *The Proc. of the 2nd Dutch/German Workshop on Nonmonotonic Reasoning*, pages 225–234, Utrecht, 1995. Delft University of Technology, University of Utrecht.
24. Nevin Lianwen Zhang and David Poole. Exploiting causal independence in bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328, 1996.
25. I. Zukerman, R. McConachy, K. B. Korb, and D. Pickett. Exploratory interaction with a bayesian argumentation system. In T. Dean, editor, *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99)*, pages 1294–1299. Morgan Kaufmann, 1999.