Book Title
Book Editors
IOS Press, 2003

Probabilistic Proof Standards for the Carneades Model of Argument

Thomas F. Gordon [a,1], and Douglas Walton [b]
[a] *Fraunhofer FOKUS, Berlin*
[b] *Dept. of Philosophy, University of Winnipeg, Manitoba, Canada*

Abstract.

Keywords. Argument Graphs, Argument Evaluation, Probability Theory, Bayesian Networks, Qualitative Probability Theory

1. Introduction

Perhaps the most well known symbol of the legal system, which can traced back to the ancient times, is the image of a blind-folded woman with the 'scales of justice' in one hand and a sword in the other. The Romans called her 'Justitia'. The scales clearly suggest that justice requires some kind of weighing or balancing of the evidence. But just how can or should this be done? Is the balancing of evidence a purely subjective matter, or can normative standards be articulated?

In this paper we investigate the use of probability theory to extend the Carneades model of argument graphs [1] to support the weighing of evidence. Carneades is a mathematical model of argument which applies proof standards to determine the acceptability of statements on an issue-by-issue basis. In [1], three proof standards for Carneades have been formally defined:

Scintilla of Evidence (SE). A statement meets this standard iff it is supported by at least one defensible pro argument.

Best Argument (BA). A statement meets this standard iff it is supported by at least one defensible pro argument with priority over all defensible con arguments.

Dialectical Validity (DV). A statement meets this standard iff it is supported by at least one defensible pro argument and none of its con arguments are defensible.

The BA standard comes closest to weighing evidence, if we model evidence as arguments. The BA standard uses a priority ordering on arguments to favor the side (pro or con) with the highest priority argument. But since arguments are aggregated by the BA standard using only an ordinal scale, it does not allow several weak arguments to be combined to defeat some argument which is stronger than each of them separately.

[1]Correspondence to: Dr Thomas F. Gordon, Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, Berlin, Germany. E-mail: thomas.gordon@fokus.fraunhofer.de

Probability theory provides quantitative methods for combining evidence which might be useful for solving this problem [2, pp. 462–648].

In this paper we analyze the Carneades model of argument from the perspective of probability theory, with the aim of developing well-founded probabilistic proof standards for Carneades allowing arguments from evidence to be weighed.

The rest of this paper is organized as follows. Section 2 presents the Carneades model of argument. Section 3 introduces probability theory and analyzes Carneades from the perspective of probability theory. Section 4 defines some probabilistic proof standards for Carneades. Section 5 illustrates the use of probabilistic proof standards with examples of legal arguments about evidence. Section 6 discusses related work and Section 7 presents our conclusions.

2. The Carneades Model of Argument

We begin by defining the structure of arguments. The premises and the conclusion of arguments are statements about the world, whether empirical or institutional, which may be accepted or rejected. For the purpose of evaluating arguments, the syntax of statements is not important. We only require the ability to determine whether two statements are syntactically equal and some way to denote the logical complement of a statement.

Definition 1 (Statements) *Let* $\langle \text{statement}, =, \text{complement} \rangle$ *be a structure, where* statement *denotes the set of declarative sentences in some language,* $=$ *is an equality relation, modeled as a function of type* $\text{statement} \times \text{statement} \rightarrow \text{boolean}$*, and* complement *is a function of type* $\text{statement} \rightarrow \text{statement}$ *mapping a statement to its logical complement. If* s *is a statement, the complement of* s *is denoted* $\overline{s}$*.*

Next, to support defeasible argumentation and allow the burden of proof to be distributed, we distinguish several types of premises.

Definition 2 (Premises) *Let* premise *denote the set of premises. There are the following types of premises:*

1. *If* s *is a* statement*, then* $\diamond s$*, called an* ordinary premise*, is a premise.*
2. *If* s *is a* statement*, then* $\bullet s$*, called an* assumption*, is a premise.*
3. *If* s *is a* statement*, then* $\circ s$*, called an* exception*, is a premise.*
4. *Nothing else is a premise.*

Now we are ready to define the structure of arguments.

Definition 3 (Arguments) *An* argument *is a tuple* $\langle c, d, p \rangle$*, where* c *is a* statement*,* $d \in \{\text{pro}, \text{con}\}$ *and* $p \in 2^{\text{premise}}$*. If* a *is an argument* $\langle c, d, p \rangle$*, then* $\text{conclusion}(a) = c$*,* $\text{direction}(a) = d$ *and* $\text{premises}(a) = p$*.*

Note that an argument may have an empty set of premises. One difference between arguments and inference rules is evident in the distinction between pro and con arguments. Semantically, con arguments are instances of presumptive inference rules for the negation of the conclusion; if the premises of a con argument hold, this provides reasons to reject the conclusion or, equivalently, to accept its logical complement. Our approach

abstracts completely from the syntax of the language for statements. Also, the use of both pro and con arguments is in accordance with our view of argumentation as a dialectical process. The arguments pro and con some statement need to be ordered or otherwise aggregated to resolve the conflict.

An argument graph is a kind of proof tree in that it provides a basis for explanations and justifications. The *acceptability* relation between argument graphs and statements is intended to model the sufficiency of the proof: intuitively, a statement is acceptable given the arguments if and only if the argument graph is a proof of the statement. This distinguishes the acceptability relation from the defeasible consequence relation of non-monotonic logics. In a calculus for such logics, assuming it is correct and complete, a statement is a defeasible consequence of a set of statements if and only if the statement is *derivable* in the calculus, whether or not such a proof has in fact been *derived*.

Argument graphs have two kinds of nodes, statement nodes and argument nodes. The edges of the graph link up the premises and conclusions of the arguments. In Carneades argument graphs, at most one statement node is allowed for every statement s and its complement, $\overline{s}$. A statement and its complement are not just two unrelated statements. A dispute about s is also a dispute about $\overline{s}$. An argument pro one is an argument con the other. If one of these statements is accepted, the other must be rejected. There are no restrictions on the use of statements in premises. Both s and $\overline{s}$ may be used in premises, no matter which statement is represented by a statement node in the argument graph. Restricting statement nodes to at most one node for each s and $\overline{s}$ pair avoids the duplication of all arguments pro s as arguments con $\overline{s}$ and helps to reduce the complexity of diagrams of the graphs. If s is represented by a statement node in an argumentation graph, then a premise using $\overline{s}$ is called a *negative premise*. Ordinary premises, assumptions and exceptions may all be negative, using $\diamond\overline{s}$, $\bullet\overline{s}$, and $\circ\overline{s}$.

In the diagrams of argument graphs which follow, statements are displayed as boxes and arguments as circles or rounded boxes. Different arrowhead shapes are used to distinguish pro and con arguments as well as the different kinds of premises. Pro arguments are indicated using ordinary arrowheads; con arguments with open arrowheads. Ordinary premises are represented as edges with no arrowheads, assumptions with closed-dot arrowheads and exceptions with open-dot arrowheads. Negative premises have an additional tee mark (a short perpendicular line). Notice that the premise type cannot be adequately represented using statement labels: since argument graphs are not restricted to trees, a statement may be used in several premises with different types.

Figure 1 illustrates this method of diagramming argument graphs with a toy legal argument. The issue is whether there is a contract. One con argument, a2, states there is not a contract because the agreement is not in writing although it is for the sale of real estate. The other con argument, a3, states the agreement is not in writing since is was by email. Argument a1 states there is a contract, because agreements are contracts unless one of the parties is a minor. Argument a5 states the agreement is for the sale of real estate, assuming there is a deed, i.e. a written instrument conveying the property. Argument a4 uses the deed as evidence of there having been an agreement. These last two arguments, especially, may not be realistic, but were contrived only to illustrate how argument graphs are not restricted to trees and how a statement can be used in different types of premises in several arguments.

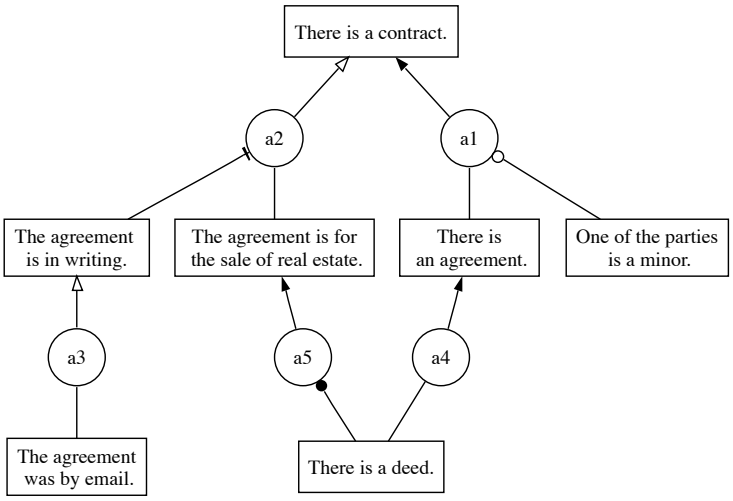

Figure 1. Argument Graph

Although argument graphs are not restricted to trees, they are not completely general; we do not allow cycles. This restriction is intended to assure the decidability of the acceptability property of statements.

Definition 4 (Argument Graphs) *An* argument-graph *is a labeled, finite, directed, acyclic, bipartite graph, consisting of* argument *nodes and* statement *nodes. The edges link the argument nodes to the statements in the premises and conclusion of each argument. At most one statement node is allowed for each statement s and its complement, $\overline{s}$.*

By argument evaluation we mean determining whether a statement is *acceptable* in an argument graph. Intuitively, a statement is acceptable if a decision to accept the statement as true can be justified or explained given the arguments which have been put forward in the dialogue. The definition of the acceptability of statements is recursive. The acceptability of a statement depends on its *proof standard*. Whether or not a statement's proof standard is *satisfied* depends on the *defensibility* of the arguments pro and con this statement. The defensibility of an argument depends on whether or not its premises *hold*. Finally, we end up where we began: whether or not a premise holds can depend on whether or not the premise's statement is acceptable. Since the definitions are recursive, we cannot avoid making forward references to functions which will be defined later.

To evaluate a set of arguments in an argument graph, we require some additional information. Firstly, we need to know the current dialectical *status* of each statement in the dialogue, i.e. whether it is stated, questioned, accepted, or rejected. This status information is pragmatic; the status of statements is set by speech acts in the dialogue, such as asking a question, putting forward an argument or making a decision. Secondly, we assume that a proof standard has been assigned to each statement. In the following, let proof-standard be an enumeration of some proof standards. Finally, we assume a strict partial ordering on arguments, which we will denote with $>$. Let $a1$ and $a2$ be arguments. If $a1 > a2$ we say that $a1$ *has priority over* $a2$. Let us formalize these requirements by postulating an *argument context* as follows.

Definition 5 (Argument Context) *Let $\mathcal{C}$, the argument context, be a tuple* $\langle \text{status}, \text{ps}, > \rangle$, *where* status *is a function of type* $\text{statement} \rightarrow \{\text{stated}, \text{questioned}, \text{accepted}, \text{rejected}\}$, ps *is a function of type* $\text{statement} \rightarrow \text{proof-standard}$ *and* $>$ *is a strict partial ordering on arguments.*[1] *For every statement s and its complement $\overline{s}$, the proof standard assigned to $\overline{s}$ is the complement of the proof standard assigned to s and*

- *if* $\text{status}(s) = \text{stated}$ *then* $\text{status}(\overline{s}) = \text{stated}$,
- *if* $\text{status}(s) = \text{questioned}$ *then* $\text{status}(\overline{s}) = \text{questioned}$,
- *if* $\text{status}(s) = \text{accepted}$ *then* $\text{status}(\overline{s}) = \text{rejected}$, *and*
- *if* $\text{status}(s) = \text{rejected}$ *then* $\text{status}(\overline{s}) = \text{accepted}$.

The constraints on the status of statements in this definition serve two purposes: First, they assure that whenever a statement is stated or questioned its complement is also, implicitly, stated or questioned. Stating or questioning a statement does imply the assertion of some position or viewpoint pro or con the statement. Stating a statement merely introduces it into the dialogue. Questioning a statement merely makes an issue out of the statement. Second, the constraints assure that whenever a statement is accepted its complement is rejected and vice versa. A decision to accept one is simultaneously a decision to reject the alternative.

In the definitions which follow, an argument context is presumed.

Definition 6 (Acceptability of Statements) *Let* acceptable *be a function of type* $\text{statement} \times \text{argument-graph} \rightarrow \text{boolean}$. *A statement s is* acceptable *in an argument graph G if and only if it satisfies its proof standard:*
$\text{acceptable}(s, G) = \text{satisfies}(s, \text{ps}(s), G)$.

Definition 7 (Satisfaction of Proof Standards) *A proof standard is a function of type* $\text{statement} \times \text{argument-graph} \rightarrow \text{boolean}$. *A statement s is* satisfied *by a proof standard f in an argument graph G if and only if $f(s, G)$ is true.*

The SE, BA and DV proof standards are defined in the introduction of this paper. Each of these proof standards depend on a determination of the *defensiblity* of arguments, defined next.

Definition 8 (Defensibility of Arguments) *Let* defensible *be a function of type* $\text{argument} \times \text{argument-graph} \rightarrow \text{boolean}$. *An argument α is defensible in an argument graph G if and only if all of its premises* hold *in the argument graph:* $\text{defensible}(\alpha, G) = \text{all}(\lambda p.\, \text{holds}(p, G))(\text{premises}\, \alpha)$.[2]

Now we come to the last definition required for evaluating arguments, of the holds predicate. This is where the dialectical status of a statement and the type of premises come into play.

[1] Should the need arise to make the proof standard of a statement depend on the argument in which the statement is used as a premise, the ps function of the context could be extended to be of type $\text{statement} \times \text{argument} \rightarrow \text{proof-standard}$.

[2] Here 'all' is a higher-order function, not a quantifier, applied to an anonymous function, represented with λ, as in lambda calculus.

Definition 9 (Holding of Premises) *Let* holds *be a function of type* premise $\times$ argument-graph $\rightarrow$ boolean. *Whether or not a premise holds depends on its type. Thus, there are the following cases:*

If p *is an ordinary premise,* $\diamond s$*, then*

$$\text{holds}(p, G) = \begin{cases} acceptable(s, G) & \textit{if } \text{status}(s) = \text{stated} \\ acceptable(s, G) & \textit{if } \text{status}(s) = \text{questioned} \\ true & \textit{if } \text{status}(s) = \text{accepted} \\ false & \textit{if } \text{status}(s) = \text{rejected} \end{cases}$$

If p *is an assumption,* $\bullet s$*, then*

$$\text{holds}(p, G) = \begin{cases} true & \textit{if } \text{status}(s) = \text{stated} \\ acceptable(s, G) & \textit{if } \text{status}(s) = \text{questioned} \\ true & \textit{if } \text{status}(s) = \text{accepted} \\ false & \textit{if } \text{status}(s) = \text{rejected} \end{cases}$$

Finally, if p *is an exception,* $\circ s$*, then*

$$\text{holds}(p, G) = \begin{cases} \neg acceptable(s, G) & \textit{if } \text{status}(s) = \text{stated} \\ \neg acceptable(s, G) & \textit{if } \text{status}(s) = \text{questioned} \\ false & \textit{if } \text{status}(s) = \text{accepted} \\ true & \textit{if } \text{status}(s) = \text{rejected} \end{cases}$$

It can be proven that there is always a unique and complete assignment of 'acceptable' or 'not acceptable' to statements, and of 'holds' or 'not holds' to premises.

Theorem 1 *For every argument context* $\mathcal{C}$ *with proof standards* SE*,* BA *or* DV *and every argument graph* G*:* acceptable *and* holds *are total functions.*

Moreover, if the BA or DV proof standard is assigned to a statement in the context, it can never be the case that both the statement and its complement are acceptable.

Theorem 2 *For every argument context* $\mathcal{C}$ *with proof standards* BA *or* DV *and every argument graph* G*, no statement* s *exists such that* s *and* $\overline{s}$ *are both acceptable.*

The proofs of the these theorems can be found in [1].

3. A Probabilistic Analysis of Carneades

Since probability theory is well known, there is no need to provide an introduction here. For an introduction from the perspective of Artificial Intelligence, see [2, pp. 462–648]. Our focus in this section is on trying to interpret Carneades in terms of probability theory.

Degrees of belief are applied in probability theory to propositions. *Random variables* are used to partition the domain of propositions. Each variable represents a set of propo-

sitions, only one of which may be true in any model. Carneades statements can be interpreted as values of Boolean random variables. For any statement s and its complement $\overline{s}$ there is a Boolean random variable with these two values. We will denote variables which range over statements with lowercase letters and use capital letters to denote the corresponding random variable. Thus the random variable S has the domain $\{s, \overline{s}\}$.

A probabilistic model, i.e. knowledge base, of some domain is constructed by declaring a set of random variables for the relevant features of the domain and assigning probabilities, i.e. real numbers in the range of 0 to 1, to every possible combination of values of these variables, i.e. states of the world. If only Boolean variables are used, then given n variables there are 2^n possible states. The assignment of probabilities to these states is called a *full joint distribution*. To be consistent with the axioms of probability theory, the sum of the probabilities of a full joint distribution must be equal to 1. This expresses that exactly one of these states must be true, even though, due to incomplete or uncertain information, we do not know with certainty which of these states is the true one.

Given a knowledge base represented by a full joint distribution, the axioms of probability theory can be used to derive the conditional probability of any proposition, including compound propositions constructed using the negation, conjunction and disjunction operators of propositional logic. In AI, Baye's theorem is often used to update the belief's of an agent in the light of new evidence:

Definition 10 *Bayes' Theorem*

$$P(p|a) = \frac{P(a|b)P(b)}{P(a)}$$

In plain text, Bayes' Theorem tells us that the probability of p given a is equal to the product of the probability of a given b and the prior, i.e. unconditional, probability of b, divided by the prior probability of a.

Probability theory per se does not tells us how to construct the knowledge base, i.e. the full joint distribution, for some domain. There are at least two problems to deal with:

1. Where do the probabilities come from? Ideally we would have sufficient empirical data from well-designed experiments. Sometimes the numbers can be deduced from a deep causal model of the domain. Neither of these alternatives are available for most practical application scenarios, including typical legal applications. The probabilities may also be asserted by one or more experts, as a formalization of their expert knowledge. This *subjectivist* approach is controversial, but has its adherents [2, p. 472]. It seems to me no more (or less) problematical, in principal, than other forms of modeling expert knowledge, for example using knowledge representation languages based on first-order logic.
2. How can domains with a large number of random variables be modeled, given the fact that the size of the full joint distribution grows exponentially with the size of the number of random variables? Suppose the domain can be modeled with 20 Boolean random variables. Even this modest-sized domain would require the experts to be able to assign probabilities to 1048576 states and do so in a way which does not violate the axioms of probability theory. This is another version of the familiar 'knowledge acquisition bottleneck'.

The later, knowledge acquisition, problem can be overcome, or at least lessened, by making use of knowledge about dependencies between random variables in the domain. This is where *Bayesian Networks* can play a role. A Bayesian network is an acyclic directed graph (DAG) representation of dependencies between random variables. Given such a network, the full joint distribution can be provided by assigning a *conditional probability table* (CPT) to each node in the graph, reducing the complexity of assigning probabilities in a domain with n Boolean variables from the exponential 2^n to the more manageable $n2^k$ numbers, where k is the maximum number of parent nodes for some variable in the Bayesian Network.

The key idea for interpreting Carneades argument graphs probabilistically is to try to find a way to map argument graphs to Bayesian Networks. Some surface similarities are immediately apparent. Both are acyclic graphs consisting of nodes representing Boolean variables.[3] Carneades has, in addition, argument nodes, but these can be ignored for the purpose of identifying variable dependencies.

4. Probabilistic Proof Standards for Carneades

5. Examples in the Domain of Legal Reasoning About Evidence

6. Related Work

7. Conclusion

References

[1] T. F. Gordon, H. Prakken, and D. Walton. The Carneades model of argument and burden of proof. *Artificial Intelligence*, 2007. In Press.

[2] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Series in Artificial Intelligence. Prentice Hall, second edition, 2003.

[3] Only Boolean random variables are relevant for our purposes here, since Carneades statements have only two values, acceptable or not acceptable.