NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Probabilistic Argumentation

by

Mohammad Sharique Zaman

School of Computer Science and Engineering

April 2018

SCE17-0268
Probabilistic Argumentation

by

Mohammad Sharique Zaman

Submitted to the School of Computer Science and Engineering, in partial fulfillment of the requirements of Bachelor of Engineering (B.Eng.) in Computer Science at Nanyang Technological University, Singapore

School of Computer Science and Engineering

April 2018

Abstract

Bayesian Networks are systems that allows probabilistic computation of argument nodes but are not capable of providing useful logical explanations for those decisions. In contrast, Argumentation deals with reasoning with arguments towards a conclusion, but does not have a way of weighing up different arguments due to a lack of a quantitative metric.

In this paper, we develop a Hybrid Argumentation Framework that allows for both probabilistic computation as well as logical explanation. We first derive an argument framework from a given Bayesian Network. We then define various semantics for determining admissibility of arguments and provide algorithm for computation of the same. After this, we identify several different forms of explanations and we provide algorithms for computing those explanations. The set of explanation forms identified is a subset of all possible explanation forms. We finally bring this all together through an algorithm, which is then implemented in a software application.

The results obtained are encouraging as explanations of the forms identified were successfully computed, but there is currently no system of ranking these explanations. Future work can focus on developing a system of evaluating these, as well as incorporating other forms of explanation that have not been covered in this system.

Acknowledgements

As I conclude my report, I am overwhelmed by an immense sense of gratitude towards a number of people whose silent contributions and support were absolutely critical in the progress and completion of this project.

In particular, I would like to thank four very special people.

First of all, my supervisor Prof Miao Chun Yan, for being forbearing where warranted but also strict when needed, keeping me on my toes but making sure I didn’t fall. Without her support and encouragement, I can’t imagine myself finishing this task.

Secondly, to Mr. Tan Benny, for being constantly at hand to assist me with my troubles and patiently answering whatever queries I had. I always felt secure knowing he had my back.

Thirdly, to my dear friend Prabhjot Vicky Grewal, for drawing on his keen grasp of argumentation to help me out whenever I was stuck at a dead end.

Fourthly, to my friend Virat Krishan Chopra for constantly pushing me to never settle for anything short of excellence and inspiring me through his example.

Lastly, I want to thank my parents for their unconditional love and emotional support during an unbelievably trying final year.

Table of Contents

Abbreviations

BAF	Bayesian Argument Framework
DAF	Dung's Argument Framework
DAG	Directed Acyclic Graph
BN	Bayesian Network

Chapter 1: Introduction

Artificial Intelligence is a booming field in Computer Science, making rapid advancements in various applications. It is no longer limited to games such as chess or go, where it has far surpassed humans, but is also making inroads into spheres which impact the lives of humans more directly in extremely important ways. Artificial Intelligence Systems are being developed in the medical sphere for disease diagnosis and setting out methods of treatment. Self-driving cars are all the rage these days, with the promise of eliminating the need for human drivers and enormous reduction in road traffic accidents. The legal profession is also anticipating a swift introduction of Artificial Intelligence to decide court cases. With this technology playing such a central role in all aspects of our lives, it becomes extremely important to understand it well.

One of the important disciplines that have been created in response to this rapid AI growth is the field of Explainable AI [1]. Computers make decisions based on rapid computations, which human beings cannot expect to follow. However, one of the endeavors of Explainable AI is to still present human-understandable reasons for the actions of AI. For example, it is not enough for a machine learning system to identify cats correctly, but we also desire it to tell us why it identified a certain object as a cat, for instance, because it has fur and whiskers and it mewed when under observation.

There are two main reasons why explainability is important. The first is to enhance human knowledge. If we are able to understand the reasons behind the decisions computer takes, that knowledge contributes to the theory in that domain. For instance, if the computer finds a correlation between symptom X and disease Y, and on the basis of that it diagnoses disease Y, then if it is also able to tell us that it's diagnosis was based on the presence of X, we can add that correlation to our knowledge base, and future diagnoses can be more accurate.

The second reason why explainability is important is safety. If a self-driving car detects humans correctly a million times, that does not completely ensure that it is safe, if the reason for its detection of humans is based on the dress the human was wearing. If the driving location changes, and with that the dress too, the car will fail. Therefore, we need to know the reasons behind AI's actions to determine whether those reasons are correct, to increase our belief in its safety.

One of the most common tasks of artificial intelligence systems is probabilistic computation within Bayesian Networks. Because these networks are extremely complicated, no trivial explanations for these computations are available. On the other hand, the field of argumentation focuses on making decisions based on clearly defined reasons through an evaluation of all the different reasons / arguments. However, these arguments are qualitative, and there is limited ability to weigh up different arguments.

These two techniques can be combined to create a Hybrid System, which allows for both probabilistic computation and logical explanation. That is the task this paper endeavors to do. The paper is organized as follows - Chapter 2 covers Literature Review of Argumentation, Bayesian Networks, and Explanations. We develop our system in Chapter 3, starting with a conversion from Bayesian Network to Argumentation Framework, and then going on to define semantics of admissibility and explanations within these systems. In Chapter 4, we explain our implementation of a software application based on this system. Chapter 5 discusses the implications of this work, its limitations, as well as directions for future work.

Chapter 2: Literature Review

2.1 Argumentation

Argumentation is the field dealing with modelling arguments, their relations with each other and evaluating arguments according to different metrics. Many different systems of argumentation exist and are used [2]. The foundational work in the field of argumentation in Artificial Intelligence is Phan Minh Dung's paper on Argumentation [3], where he defines an argument framework as follows:

Definition 1: Argument Framework

An argument framework is a pair <A, R> where A is the set of arguments, and R is a binary relation on A called attack relation.
We'll represent an attack relation between arguments a and b where a attacks b as a -> b.

Most other works on argumentation either use this definition of argument framework or build upon it. This is because it is sufficiently general as it does not specify any form of argument or their attacks.

Since no quantitative data is available on arguments under Dung's framework, individual arguments cannot be assigned any degree of belief. Instead, argument evaluation under Dung's Framework amount to extracting subsets of arguments, called extensions which are collectively considered justified.

The function that computes these sets is called an argumentation semantic. Based on need, many different semantic [4]s have been proposed, some of which are admissible, preferred, grounded, CF-2, etc. All of these impose different conditions and constraints on the extensions. These constraints have not been explored here have not been explored here as they are not relevant to this work.

Most semantics satisfy at least two conditions -

a) They are consistent with each other, i.e. they do not attack each other
b) If an argument outside the extension attacks an argument within it, another argument within the extension must attack the former. This can be looked at as protecting arguments within the extension from external arguments.

2.2 Bayesian Networks

A Bayesian Network [5] is a probabilistic model which represents a set of variables through nodes in a Directed Acyclic Graph (DAG) and the conditional dependencies between those variables through edges. The quantitative values of those conditional dependencies are contained in Conditional Probability Tables (CPT). Each node in a Bayesian network has a CPT associated with it and may have multiple states. Leaves of the Bayesian Network, i.e. nodes having no parents have a priori probabilities associated with them.

Consider the simple Bayesian Network with three nodes, rain, sprinkler and wet grass, with the former two connected to the latter. All three nodes have two states, (True, False). The conditional probability table for wet grass might look like this

Rain	**Sprinkler**	**Wet Grass(True)**
True	True	0.99
True	False	0.60
False	True	0.30
False	False	0.10

Let's interpret the second row. We say, P (Wet Grass (True) | Rain (True) ^ Sprinkler (False)) = 0.60. This says that if it rains but the sprinkler is not used, the probability of the grass being wet is 0.60.

Another notion in Bayesian Networks is the notion of Joint Probability. We define it here as we will be using it later as a metric for argument strength -

Definition 2: Joint Probability

Joint probability of a number of events is the probability of all those events occurring at the same time.

In the system we develop to utilize a Bayesian Network for computation, two inputs need to be provided.

1) A set of evidence nodes. These are nodes whose state is known from a specific observation, i.e. assigned a probability 1.

2) A query node and its desired state. This is the node for which the probability for it being in the desired state is required.

With these definitions, we are ready to use Bayesian Networks in our system.

2.3 Explanation

There is no commonly accepted definition of what constitutes an explanation, and it is often very domain-specific. In Artificial intelligence, in a loose sense, any presentation that helps the human user understand the reasoning behind a particular action taken by an autonomous agent can be considered an explanation [6]. However, increasingly, the most important form of explanation being required in artificial intelligence systems is abductive inference. Abductive inference is a form of logical inference which starts with an observation or set of observations and then seeks to find the simplest and most likely explanation. For instance, in the case of a self-driving car, the observation may be that the care braked at a particular point. The attempt to find the best explanation for this behavior of the car will be abductive inference [7]. In Bayesian Networks also, we are concerned with adductive inference, where we observe a certain state of a node, and seek to find the best explanation for why that state was observed. In Bayesian systems, each explanation consists of a particular configuration of the nodes, which represents the following logic ‘because these other nodes had these states, the observed node had the said state’. For example, raining(true) is one explanation for wet grass(true).

With a firm understanding of Argumentation, Bayesian Networks, and Explanations, we are ready to build our Hybrid Argument Framework, which we will call the Bayesian Argument Framework (BAF).

Chapter 3: Developing the Bayesian Argument Framework, Semantics and Explanations

In this section, we will first of all set down the method of obtaining an argument framework from the Bayesian Network. We will then define 4 different semantics for argument admissibility in this framework and develop algorithms for computing them. We'll then develop our notion of explanation in this framework and algorithms for computing them. We'll finally put this all together in an algorithm for probabilistic computation and logical explanation of a query node, given a Bayesian Argument Framework.

3.1 Developing BAF

3.1.1 Extracting a rule base from Conditional Probability Table

The prerequisite to any argumentation process is having a set of arguments, so that attack relations among them can be identified, and argument evaluation can be performed. The simplest representation of an argument is a rule of inference, i.e. *if a certain condition X is true, Conclusion Y will occur.* One incidental advantage of working with Bayesian Networks is that rules such as these can be gleaned very easily from the knowledge base [8], as the information in the Bayesian Network is in the form of causal relationships between different variables which can be converted into such rules. Since CPT s store all the information regarding the causal relationships, all that is required is to convert the CPTs as well as the evidence set into the rule-base and fact-base respectively.

One important thing to note regarding this conversion is that for every CPT, rules have been obtained from all marginal probability tables for that CPT. For example, for the CPT for rain and sprinkler as parents and wet grass as the child, we'll obtain 'rain => wet grass', 'sprinkler => wet grass' and also 'rain ^ sprinkler => wet grass'. This is necessary to ensure that after argument evaluation, the probabilities due to a particular fundamental condition (causal node) are not diluted due to being split among irrelevant attached conditions.

3.1.2 Conversion of Rule Base into Arguments

The rule base obtained here now needs to be converted into arguments. This conversion cannot be performed directly by mapping each rule to an argument in the argument set. This problem arises as the simple Dung's Argumentation Framework no longer suffices for this case. This is because Dung's framework only allows all information to be captured in a set of distinct arguments and another set of their attack relations. However, in this case, we also need to account for arguments that support other arguments, as that is a fundamental aspect of Bayesian Networks, where chains of causal connections between nodes are common, which are essentially conditions that support the existence of other conditions.

For that reason, we need to extend on Dung's framework, and include provision for argument supports. For that purpose, we will store the arguments as an association between a set of premises and a conclusion, and we'll define the notion of a sub-argument as follows -

Definition 3: Premise

The causal condition in a causal rule of inference. Traditionally, the LHS of the rule.

Definition 4: Conclusion

The effect of the condition in a causal rule of inference. Traditionally, the RHS of the rule.

Definition 5: Sub-Argument

An argument x' is the sub-argument of argument x, if the conclusion of x' is the premise of x, or if x' is a sub-argument of a sub-argument of x.

Note that sub-arguments can be at different levels, which is why that last condition is added to the definition of a sub-argument.

With these definitions in hand, we can chain the rules from the rule base into trees of arguments and sub-arguments. The roots of these trees will be the different "effects", or the RHS from the rule base, and the LHS of those rules will form the different branches of those trees. The leaves of those trees will be the nodes for which no causes are found, which will essentially be the leaves of the Bayesian network themselves.

One last step that needs to be taken to make this transition complete is to deal with evidence. Once an evidence is obtained, further proof of that through arguments is unnecessary. Therefore, the argument tree should be pruned from below the evidence nodes in the tree, i.e. all sub-arguments of that evidence should be removed.

3.1.3 Attacks

Now that we have a set of arguments, we need a set of attack relations between them to obtain a complete argumentation framework. I discuss below how we can attempt to obtain attacks from Bayesian networks.

There are broadly three main ways that arguments attack each other [9]. The first is called 'rebuttal attack', where an argument directly challenges the conclusion of another argument it attacks. For instance, the argument 'John is a good guy as he is obedient' is attacked by the argument that says, 'John is a bad guy because he tells lies'. This is an instance of rebuttal attack, where the latter argument denies the conclusion of the former.

The second kind of attack between arguments is the 'assumption attack'. In this case, the attacker challenges the assumption or the premises of the argument it attacks. For instance, John is not obedient will be an assumption attack to the argument mentioned above.

The final form of attack between arguments is called 'undercutting attack'. An undercutting argument is one which denies the underlying logic or the rule of inference of the argument it attacks. For instance, an argument which says, 'Being obedient does not make you a good person' challenges the underlying rule of the first argument, i.e. 'obedience makes you a good person'.

In the case of Bayesian Networks, we are interested only in the first two kind of attacks. This is because the underlying rules of inference correspond to the causal relations of the Bayesian networks, which are assumed to be true as we assume the Bayesian network to be correct.

Rebuttal attacks in the Bayesian networks are the arguments that deny the conclusion of other arguments, for instance, an argument concluding 'weather(Rainy)' is attacked by the argument '~weather(Rainy). However, there is no notion of negation in Bayesian arguments. Therefore, arguments that deny other arguments are essentially arguments that conclude in a different state of a node(s) than the arguments they attack. Such an example would be an argument with the conclusion ‘weather(Sunny)’.

Assumption attacks in the Bayesian networks are the arguments that attack the underlying premises, i.e. the causal conditions or the sub-arguments of the arguments they attack. For example, ‘weather(Sunny) ^ raining(true) => humid(true)’ is attacked by ‘raining(false)’.

Definition 6 Rebuttal Attack:

A rebuttal attack to an argument is an argument that concludes with a different state of the conclusion node as that argument.

Definition 7 Assumption Attack:

An assumption attack to an argument is an argument that concludes with a different state of any sub-argument node in that argument.

Attempted Attack vs Dominating Attack

We need to consider one more factor before defining attacks. There are two possibilities when defining attack. The first is to only consider all attempted attacks, regardless of how believable they are, while the second is to have some threshold of believability before we include an attempted attack in the attack set. A review of the literature shows that argument attacks in most standard semantics are significant in the sense that they actually weaken the original claim significantly. For example, in Dung’s admissible semantics, one of the requirements for an extension to be admissible, if an argument of the extension is attacked by an argument outside the extension, the latter must be attacked by another argument of the extension. This suggests that the attack made the argument unjustified unless the attacker was attacked itself, i.e. the attack wasn’t merely an attempted attack, but it was an attack which could dominate the original argument.

For reasons of consistency with the literature, we also use this interpretation of an attack as an attack, one which dominates the original argument. To adjudicate whether or not an argument dominates another, we need a measure of argument strength.

Argument Strength

There are two components of any argument. The premises of the argument, and the rules of inference that lead from premises to the conclusion. To adjudicate the strength of an argument, one must consider both of these. However, in the current exercise of obtaining an argumentation framework from a given Bayesian Network, the rules of inference correspond to the causal relations between nodes in the Bayesian Network, which are assumed to be correct. Therefore, we are only interested in the likelihood of the premises being true.

It follows then in this context that an argument strength is the likelihood of all of its premises, or the nodes that form the argument to be true. This is essentially the join probability of argument nodes. We need to exclude the probability of the conclusion node in this joint probability calculation, the reason being that we are not comparing the conclusions but instead the strength of the argument, i.e. the chain of reasoning that leads to the conclusion. For instance, lets suppose we have a prior belief that the state of a conclusion node is true. We still wish to be able to compare the arguments for and against the state of that node being true. If we were to consider the probability of the conclusion node however, the argument concluding in the state being true will always 'win', and we won't be able to compare the strength of the chain of reasoning. Therefore, we need to exclude the conclusion node's probability from the measure of argument strength.

Formally,

Definition 8 Argument Strength:

Argument strength is the joint probability of all argument nodes, except the conclusion node.

We can now define attack formally as follows -

Definition 9 Attack:

An argument a attacks another argument b if it satisfies the following two conditions.

a) a is either a rebuttal attack or an assumption attack to b.

b) a is stronger (has greater argument strength) than b.

We can now formally define Bayesian Argumentation Framework as follows -

Definition 10 Bayesian Argumentation Framework (BAF)

A Bayesian Argumentation Framework is a pair <A, R> where A is a set of argument-sub-argument chains, each associated with conditional probabilities, while R is a set of attack relations over these arguments.

3.2 Semantics

3.2.1 Comparison between semantics in quantitative systems vs non-quantitative systems

Now that we have an argumentation framework, we turn to its semantics to evaluate justified arguments.

Semantics as discussed earlier are operations performed on argumentation frameworks that compute sets of extensions or justified arguments. Admissible Semantics, Preferred Semantics, Grounded Semantics, etc. are some examples of such semantics.

However, this definition of semantics in Dung's Argument Framework, operates under a constraint that we do not have in BAFs. This constraint is the lack of quantitative ways of comparing different arguments. That is why we need to settle for computing sets of admissible extensions, or essentially different consistent world-views, rather than the most likely or the most correct worldview.

For example, consider the three arguments A: 'John is good as he is obedient', B: 'John is not obedient as he doesn't follow orders', C: 'John follows orders'. In this example, B attacks A, and C Attacks B.

The two consistent sets of arguments are <A, C> and <B>. However, we do not have any means of choosing between these sets, as either of those could be true. Therefore, the semantic evaluation in argument frameworks without quantitative belief data stops at the point of computing different possible extensions.

However, suppose the agent has a strong prior belief (99% certainty) that John follows orders. That immediately raises the likelihood of the first set being the true argument set. In this way, with quantitative data, we get more certain and informative answers or results of argumentation.

Therefore, since we do have quantitative data, we do not need to compute extensions. Rather, we use our semantic function to evaluate individual arguments and determine their admissibility.

3.2.2 Notion of Defeat

It follows from the previous discussion that the semantics used in BAF will be very different from the ones in the non-quantitative systems such as DAF. We still survey the traditional DAF semantics to serve as a guide when determining criteria for admissibility.

The two requirements that are common in all Dung's semantics are conflict-freeness and self-defense, i.e. if an argument outside the extension attacks an argument within the extension, there must be another argument within the extension attacking the former.

Since we do not have sets of arguments, conflict-freeness is not relevant. However, self-defense deals with the notion of attack, which is still relevant in our case. What DAF semantics essentially requires vis-a-vis acceptability is for arguments to not be dominated or defeated by other arguments.

We can borrow the same notion to Bayesian Argument Frameworks. We define the notion of defeat as follows -

Definition 11 Defeat: A defeated argument is one which is attacked by an undefeated argument. An argument which has not been defeated is undefeated [10].

Notice in this case, we attack refers to a dominating attack, i.e. an attack by an argument stronger than the argument itself, not an attempted attack.

The following algorithm computes whether an argument is defeated or undefeated

Algorithm 1-

Require: A set S of arguments with conditional probabilities with a flag set initially to undecided, and a set R of attack relations among arguments, and a specified argument A whose defeat status we are interested in finding.

```
1: M: = arguments that attack A
2: for each K ∈ M do:
3:      if K is undecided:
4:           determine K's defeat status by running it through the subroutine
5:           starting line 1
6:      if K is undefeated:
7:           set A to defeated:
8:           end for
9:      else if K is defeated:
10:          continue
11: return A's defeat flag
```

Notice that the argument will not run into an infinite loop. That can only happen if one of K's attackers is A itself, in which case it will go back and forth. However, that is not possible, as K only defeats A if it is stronger than A. If that is true, then A will be weaker than and therefore cannot attack K. Thus, there will be no infinite loop in this algorithm.

3.2.3 Proposed Semantics

3.2.3.1 Semantic 1

Under semantic 1, given a query node and a desired state of that node, the set of admissible arguments for that node are the ones that satisfy the following two conditions -

1) The arguments conclude in the desired state of the query node.
2) The argument is undefeated.

Semantic 1 can be seen as a competition-based semantics, in that all it cares about is the result of the computation of attacks between arguments, returning the arguments that survive. It does not consider the absolute strength of the arguments, only the relative strength. The most plausible arguments are the ones that are not attacked by a stronger argument and have the maximum argument strength. Such a semantic is useful in cases such as medical diagnosis where often the user has to choose between several competing options, to decide the most likely one. Another feature of these cases is that the probability used represents the likelihood of certain scenarios occurring, and therefore the quantitative metric of argument strength strongly overlaps with the actual likelihood of certain events which generally we are concerned with. However, there are some other cases where these conditions are not satisfied, and we need another criteria in the semantic to account for them.

3.2.3.2 Semantic 2

Motivation:

In many real-world scenarios, such as voting for a candidate in elections, or making a judgement in a criminal case, the agents are not just concerned with evaluating arguments with respect to each other, but also finding the arguments that are plausible enough in and of themselves to be used as a basis for some decision. For instance, it may be the case that of the two candidates in an election, one is slightly more persuasive than the other because their lies are less obvious, but that is not always a sufficient condition to vote for that candidate. Apart from the need for an absolute metric for evaluation, another factor that necessitates another semantic is the fact that quantitative differences in probability don't always strongly correlate with believability in certain arguments. For instance, just because the joint probability of a certain argument is more than the other does not make it more persuasive in the cases where the likelihood of the argument does not have a real meaning and what we are concerned with is simply whether we believe the argument to be true or not. These cases might not represent the best use of Bayesian networks, as those are best utilized in cases where probabilities of events represent likelihood. However, it may still be a useful approximation to evaluate different competing arguments.

A third reason is that sometimes, agents are not completely consistent in their worldview, and it may still be useful to model them as such. For example, a certain agent might be persuaded by two arguments that lead to different conclusions, or may be persuaded by an argument, whose premise has been attacked effectively by a stronger argument, simply because the agent believes both to be true.

With this in mind, we propose a second semantic, defined as follows:

Semantic 2: Under semantic 2, given a query node and its desired state, the set of admissible arguments for that node are the ones that satisfy the following two conditions -

3) The arguments conclude in the desired state of the query node.
4) The conditional probability of all argument nodes (the argument and all its sub-arguments) is greater or equal to 0.5

Once such a set is obtained, the ranking among the arguments is done on the basis of argument strength.

The 0.5 probability represents the agent's belief that it is more likely that the argument is true than false. For an agent to believe that an argument is overall true, they need to be convinced by the argument. If the agent does not have a strong belief in any premise of the argument, then the argument will not work, as the agent cannot be sure to a reasonable degree of certainty that the argument is correct. Hence this requirement of conditional probability being greater than 0.5 has been placed at every node of the argument.

A good real-world example of where this kind of semantic might be useful is an election debate between candidates. If a candidate makes an argument, one of the premises of which the agent believes is more likely false than it is true, they are unlikely to believe in that argument. Therefore, in such cases Semantic 2 can be employed.

3.2.3.3 Semantic 3

Motivation: So far, we have discussed cases where we are either concerned with the competition between arguments, or in the absolute strength of arguments. There might well be a scenario where the agent requires both those metrics. For instance, am election debate where candidates present quantitative data which can be easily be subjected to a likelihood analysis in the Bayesian networks, but also the agent wishes for each argument node to be more plausible than implausible. We define Semantic 3 for these cases -

Semantic 3: Under semantic 3, given a query node and its desired state, the set of admissible arguments for that node are the ones that satisfy the following two conditions -

1) The arguments conclude in the desired state of the query node.
2) The conditional probability of all argument nodes (the argument and all its sub-arguments) is greater or equal to 0.5
3) The argument is undefeated.

3.2.3.4 Semantic 4

Motivation:

There are sometimes situations where we are not just interested in the conclusion being correct but also the ‘diagnosis’ being correct. High important scenarios such as war or medical diagnosis, or conditions where further actions are dependent on not just the conclusion but the reasons for the conclusions might employ this semantic.

Semantic 4:

Under semantic 4, given a query node and its desired state, the set of admissible arguments for that node are the ones that satisfy the following two conditions -

1) The arguments conclude in the desired state of the query node.
2) The argument is undefeated.
3) The join probability of all argument nodes (including the conclusion node) is greater than or equal to 0.5.

We can notice from this discussion that under different scenarios, the semantic function required for argument evaluation may change. This is explored further in the discussion section.

3.3 Explanations

We now have both a robust argumentation framework, as well as semantics that can be used to obtain admissible arguments. We will need these to develop a system of explanations [11] for arguments.

As discussed earlier, in Bayesian networks, giving explanations amount to selecting a subset of nodes [12] through which an observed result can be best understood. Since the Bayesian network is usually heavily interconnected, observation of any particular evidence may have far reaching effects on the probabilities of all other variables, not just the immediately connected causal variables. For instance, imagine a simple Bayesian network with three nodes, rain, sprinkler and wet grass, with the former two connected to the latter in a causal connection. Since rain and sprinkler have no causal relation, it may seem that evidence on those nodes will not affect each other's probability. However, assume wet grass is true. Now, if evidence on sprinkler is observed, probability of rain will be affected, and in fact decrease, because the wet grass is now explained by sprinkler, and rain is therefore rendered less likely.

We can gather from this that the task of computing explanations is not trivial as it could take various forms. In this work, I have tackled four of these forms of explanations which will be described below. These have been chosen because these are the most recurring kinds of explanation in Bayesian networks, and also future development of explanations in Bayesian networks can be based on these works. This has been further explored in the discussion section.

3.3.1 Explanations of the first form

The first form of explanation that we use is simply the causal explanation. This will be used to explain the presence of a particular state in a node. This is the most basic kind of explanation in Bayesian networks. The causal explanation for a node with the presence of any given state or its absence will just be the presence of nodes which are linked to the node through causal chains, in particular, the parents of the given node in Bayesian networks. Explanation beyond the parents to other ancestors is a second level explanation which can be obtained by seeking explanations for the parents of the given node. For example, if sun -> heat -> sweat, we only include heat in the explanation of sweat. If the user is interested further, they could query the explanation for heat and reach sun, so we do not need to include that initially.

A causal explanation will be some combination of the parent variables, given in the node's Conditional Probability Table, which leads to a probability of the target node occurring. In the previous section, these causal relations from the Conditional Probability Table were converted into arguments. Therefore, the causal explanation for a particular node will be the arguments for that node.

It may be the case that a causal chain might have a weak link, i.e. a premise which has strong evidence against it. That causal chain can therefore not be a good explanation for the given target node. This is exactly the concept which was discussed in computing admissible extensions, making sure that the arguments are well-defended.

We can therefore see that the causal explanations for a particular node will be the set of admissible arguments which conclude in that node.

This admissible set can be computed using any of the semantics described in the previous section. Usually we are interested in the few best explanations for a particular scenario, which can be computed based on argument strength.

We now describe an algorithm to compute top k causal explanations.

==

Algorithm 2 -

Require: A query node with a desired state, a function isAdmissible that computes whether an argument is admissible or not a set A of all arguments with conditional probabilities, and a set R of attack relations over A, and k a number of top arguments required

```
M:= arguments that conclude in the query node with the desired state
R:= empty set
for each argument K in M do
    if isAdmissible(K, A, R):
        Insert K in R
sort R based on argument strength
if |R| < k
    return R
else
    return top k arguments of R
```

==

The isAdmissible function can be obtained by extending the isDefeated algorithm described previously with the specific requirements of the chosen semantic.

To illustrate, for Semantic 3, the isAdmissible function’s algorithm will be as follows -

==

Algorithm 3

Require: A set S of arguments with conditional probabilities with a flag set initially to undecided, and a set R of attack relations over arguments, and a specified argument A whose admissibility status we are interested in finding.

```
P:= arguments that attack A
M:= empty set
function allPlausible(X) =
    for each sub-argument T of X:
        if T has no sub-arguments:
            if conditional probability of T > 0.5:
                return true
        else:
            if conditional probability of T < 0.5:
                return false
            else:
                flag = 1
                for each sub-argument J of T:
                    if not allPlausible(J):
                        flag = 0
                        end for
                if flag = 1:
                    return true
                else:
                    return false
for each argument K of P:
    if allPlausible(K):
        insert K into M
```

```
24: for each argument T of M:
25:     for each sub-argument
26: for each K ∈ M do:
27:     if K is undecided:
28:             determine K's defeat status by running it through the subroutine
29:             starting line 1
30:     if K is undefeated:
31:             set A to defeated:
32:             end for
33:     else if K is defeated:
34:             continue
35: return A's defeat flag

=====================================================================
```

3.3.2 Explanations of the second form

This form of explanation explains the failure of the query, i.e. the explanation for why the query node was not found to be in the state required in the query node.

This is again a causal explanation, except, the explanation consists of the admissible arguments for states of the query node other than the one required. The implementation of this algorithm is the same as that of the first-form, with only a change in the input.

3.3.3 Explanations of the third form

We now look at the third form of explanation. In Bayesian networks, if an evidence node is observed, the probability of the parents of that node correspondingly increase. This is intuitive, since if an effect is observed, the cause is expected, as per the adage, 'there is no smoke without fire'. Illustrating with our previous example, if the grass is found to be wet, the likelihood of rain increases. In the second form of explanation, we model this scenario.

If any of the descendants of the query node is an evidence node, that constitutes an explanation for the presence of the query node. Within our argumentation framework, given a query node with a given state, if that node with a desired state is present as a premise in any argument that concludes in the evidence node with the given state, that argument is an explanation for that query node. Note that being present as a premise in the argument includes being present as a premise at any point in the sub-argument chain.

The explanation can be presented in a suitable format so as to convey to the use what form the explanation is of.

We now give an algorithm to compute explanations of the third form.

===

Algorithm 4

Require: A set S of arguments with conditional probabilities with a flag set initially to undecided, and a set R of attack relations over arguments, a set of evidence nodes E, and a query node with a given state Q whose second form explanation we seek to find

```
R:= empty set
for each evidence C in E:
    for each argument A in S:
        if the conclusion of the argument is E:
            for each node N in the traversal of the argument tree of E:
                if Q = N:
                    insert A in R
                    end for
end for
```

===

3.3.4 Explanations of the fourth form

This form of explanation explains the absence of a state in the query node in the scenario where an evidence node that was explained by the query node as a cause, is better explained with another parent or ancestors becoming known to be true [13], i.e. evidence. For example, if wet grass is caused by sprinkler and rain, and we know the grass is wet, the probabilities of rain and sprinkler will rise. However, if we know that the sprinkler was used as well, the probability of rain will decrease, as the effect of rain is now more strongly explained by the sprinkler.
We define the following algorithm to compute explanations of this form

Algorithm 5

Require: A set S of arguments with conditional probabilities with a flag set initially to undecided, and a set R of attack relations over arguments, a set of evidence nodes E, and a query node with a given state Q whose fourth form explanation we seek to find

```
R := empty set
for each evidence C in E:
    for each argument A in S:
        if the conclusion of the argument is E:
            for each node N in the traversal of the argument tree of E:
                if Q!= N: and Q ∈ E
                    insert A in R
                    end for
end for
```

3.3.4 Comparison of the first form with the rest

The first form of the explanation is the only one for which we have developed a quantitative measure of strength. This is a limitation of this system and requires future work and has been explored in the discussion section.

3.3 Combining formulations

We now have an argumentation framework, semantics for admissibility and methods for computing explanations. We now need to combine all of these to facilitate the most common use cases.

The use case for which we will tailor this system is as follows - We will have a Bayesian Network, a given query node with a particular state, and some evidence nodes. We will be required to compute

1) Probability of the query node being in the given state
2) Explanations for the same

We will use all four forms of explanations to create the explanation set.

First of all, we will create the argument chain and identify attack relations to obtain the argument framework for the application. Once we have that, we can perform the following algorithm on it to execute the tasks.

Algorithm 6

Require: A Bayesian Network BN, a set S of arguments with conditional probabilities with a flag set initially to undecided, and a set R of attack relations over arguments, a set of evidence nodes E, and a query node with a given state Q whose explanation we seek to find, a semantic definition D, a number k representing the number of top explanations of form 1 required, a probability evaluation function Pr, and functions E1 through to E4 for Explanation Forms 1 through 4(Algorithm 3, 4 and 5) respectively that return the explanation sets according to the given semantic definition D

```
Probability = Pr (BN, Q)
Ex =: empty set
for each explanation Z in E1(S, R, k, Q):
        insert Z into Ex
for each explanation Z in E2(S, R, k, Q):
        insert Z into Ex
for each explanation Z in E3(S, R, Q):
        insert Z into Ex
for each explanation Z in E4(S, R, k, Q):
        insert Z into Ex
return Probability and Ex
```

This algorithm returns the probability of the node and the different explanations for it. With this, we have the theoretical groundwork for the development of an application. We have therefore reached the end of this section.

Chapter 4: Software Implementation of the Model

In this section, we describe how the models and algorithm described here have been implemented in a software application.

4.1 Flow of the Application

First of all, the system is provided a Bayesian Network, complete with Conditional Probability Tables through a User Interface. Next, the User selects a set of evidence nodes from the network and fixes them to a particular state. The User then provides a query node, for which the probability and explanations have to be computed.

The system then begins processing this information. First of all, the set of arguments are generated from the conditional probability tables. Then, the probability of the query node, followed by the different forms of explanation for the query node are computed. These two are finally returned to the user through the interface.

4.2 Technical Details

4.2.1 Technical Stack

The algorithm has been implemented in python. Python was chosen primarily because of its readability, which is a great help when implementing a sophisticated algorithm. There a number of open source libraries and utilities which aided development as well.

The application has been packaged as a web application. The backend is a Django server, which is a high-level framework for python web development. It allows for rapid application development without needing to bother too much with the setting up and configuration of the server.

The frontend has been built with the JavaScript framework ReactJS. It’s component abstraction makes UI design very intuitive and therefore effective. The UI libraries Bootstrap, and Google’s Material UI have also been used to decorate the user interface.

4.2.2 Input Description

The Bayesian Network is described using a Genie 2.0 XDSL file, a commonly used file format to describe Bayesian Networks. The network can be generated through the user interface provided by Genie, and the resultant network can be exported as a file and fed to the program.

The evidence nodes are provided as an array, while the query node and state are provided as a string. The User Interface has input fields to make input seamless.

4.2.3 Probabilistic Computation

Instead of performing probabilistic computations manually, which would be reinventing the wheel, and also prone to mistakes, we used the java library jSMILE. jSMILE is a platform independent* library of Java classes for reasoning in graphical probabilistic models, such as Bayesian networks and influence diagrams. It can be embedded in programs that use graphical probabilistic models as their reasoning engines.

Probabilistic Computation using jSMILE is fast as it has been optimized. Essentially, whenever probabilities of a particular node are required, a function is called which in turn uses jSMILE functions to compute the probability and returns it to the calling function. To use a Java Library inside python, a JVM is started in python, and the library is run on top of it.

4.2.4 Explanation Search

We develop argument tree from the Bayesian Network and compute explanations according to the algorithms developed above. The results are passed to the frontend to display.

4.3 Experimentation and Results

The system was tested using frequently used Bayesian networks such as rain probability and medical diagnoses. The results were encouraging as they computed the probabilities and returned the explanation set where any of the different forms of explanations were found. More work is needed for more intelligible presentation of these explanations, as right now there are too many of them, which could be overwhelming for the user.

Chapter 5: Discussion

5.1 Contributions

This work takes an important step towards making Bayesian network computation intelligible. The formulations of explanations, although incomplete, show a path on which further developments can be made.

The paper has an educational contribution as well, as it thoroughly and methodically explains the development of this system which is extremely important as there is a dearth of simple introductory works in this field.

I also draw clear distinctions between semantics in non-quantitative and quantitative systems, which gives us freedom from the heavily restricted notion of extensions in non-quantitative systems such as Dung’s Argument Framework.

We finally also discuss a number of semantics and the different cases where they might be useful, which opens up potential for applicability in a variety of different cases.

5.2 Practical Usage

The algorithms developed can be used in two main scenarios. The first is the case where a Bayesian Network is available, and the user is interested in obtaining explanations for certain nodes. In that case, the system can be used directly.

However, in cases where only qualitative arguments are available, we can still model a Bayesian Network after the agent’s belief in those arguments and their causal relations. For instance, a jury member could be made to fill up a questionnaire with questions such as “On a scale of 1 to 10, with 5 being ambivalent, 0 being sure of the falsity of the argument and 10 being sure of its truthfulness, rate the following statement. “A person of good character would not commit a cold-blooded murder.” The responses could then be compiled to form conditional probability tables, from which Bayesian Network could be formed. Particularly helpful is the fact that we are not totally restricted by the Semantic 1 which assumes a certain degree of sameness between the belief in an argument and the likelihood of the argument occurring. Where that correspondence is not very strict, i.e. belief represents truthfulness of arguments but not necessarily the likelihood of it occurring, we can use other Semantics which have been proposed.

5.3 Limitations

The major limitation of the current system is that no quantitative measurement of the strength or relevance of explanations (except the explanation of first form is available).

That makes it impossible to filter the set of explanations to the strongest ones, which results in too many explanations in certain cases.

The second problem is the inability to cross compare explanations of one form with the another, as no common metric of explanation strength or relevance is available.

Finally, several forms of explanations can still not be computed. A more comprehensive system of explanations needs to be developed.

5.4 Future Work

There are three main areas on which future work can be done. The first is the development of semantics. Semantics defined in this work represent the most commonsensical use cases of Bayesian nets. However, domain specific knowledge may add or relax more requirements to these semantics. For instance, a threshold of argument strength based on previous experience, or a requirement of having a certain number of sub-arguments, etc. can all be imagined as being important part of some domain specific semantics. Further work is needed to identify most commonly used semantics as well as establish guidelines when deciding those.

Second work is to refine the computation of explanation. Other forms of explanation need to be developed. A joint strength measure to compare explanations with other explanations of the same form and of different forms is needed. Theoretical investigation into Bayesian Networks is necessary to decide whether there is a finite form of explanation forms, and if not, to investigate other ways of obtaining relevant explanations.

Thirdly, quantitative probability measures need to be equated with some more human understandable measures to help guide decisions based on these networks. For instance, how good is an argument with a joint probability of 0.08 is not immediately clear to human beings.

Comprehensive study of different Bayesian Networks is needed to identify if an approximate measure can be obtained.

5.5 Conclusion

We have developed a Hybrid Argumentation System, that combines the probabilistic computation of Bayesian Networks with Logical Reasoning of Argumentation Systems. Our method of computing explanations will be refined in future works. However, it is an important first step in making Bayesian networks explainable.

Works Cited

[1] T. W. K.-R. M. Wojciech Samek, Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models, 2017.

[2] P. E. D. T. J. M. Bench-Capon, Argumentation in Artificial Intelligence, 2007.

[3] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, 1995.

[4] M. G. Pietro Baroni, Semantics of Abstract Argument Systems.

[5] M. Horny, Bayesian Networks, 2014.

[6] G. F. C. H. J. Suermondt, An Evaluation of Explanations of Probabilistic Inference.

[7] H. L. T.-C. L. Changhe Yuan, Most Relevant Explanation in Bayesian Networks, 2011.

[8] J. W. Matt Williams, Combining Argumentation and Bayesian Nets for Breast Cancer Prognosis, 2006.

[9] D. Walton, Objections, Rebuttals and Refutations.

[10] G. A. W. Vreeswijk, Argumentation in Bayesian Belief Networks, 2004.

[11] F. J. D. Carmen Lacave, A Review of Explanation Methods for Bayesian Networks, 2002.

[12] J.-P. P. A. E. Ulf H Nielsen, Explanation Trees for Causal Bayesian Netwokrs.

[13] M. N. a. D. L. Norman Fenton, Modelling mutually exclusive causes in Bayesian networks.