On Learning Attacks in Probabilistic Abstract Argumentation

Régis Riveret
Data61, CSIRO, NICTA
Brisbane, Australia
regis.riveret@data61.csiro.au

Guido Governatori
Data61, CSIRO, NICTA
Brisbane, Australia
guido.governatori@data61.csiro.au

ABSTRACT

Probabilistic argumentation combines the quantitative uncertainty accounted by probability theory with the qualitative uncertainty captured by argumentation. In this paper, we investigate the problem of learning the structure of an argumentative graph to account for (a distribution of) labellings of a set of arguments. We consider a general abstract framework, where the structure of arguments is left unspecified, and we focus on the grounded semantics. We present, with experimental insights, an anytime algorithm evaluating 'on the fly' hypothetical attacks from the examination of an input stream of labellings.

Keywords

Probabilistic Abstract Argumentation; Structure Learning.

1. INTRODUCTION

The combination of formal argumentation and probability theory to account for uncertainty has been given increasing attention in recent years, in particular with regard to abstract frameworks, see e.g. [12, 21, 2, 6, 15, 19, 20]. In this context, non-trivial problems regard (i) the (efficient) computation of the probability of arguments' statuses given a background argumentative knowledge (and with the assumption that arguments are not probabilistically independent), and (ii) learning the probability distribution of arguments' statuses from examples of argument's statuses. Another problem, which has not been addressed so far and that we call here *abstract structure learning*, concerns the induction of an argumentative structure accounting for arguments' statuses drawn from an unknown probability distribution. This paper addresses this problem, by presenting an anytime algorithm evaluating 'on the fly' hypothetical attack relations amongst arguments from the examination of a sequence of arguments' statuses. We consider a probabilistic abstract framework, leaving thus the possibility to complement this work with techniques for learning the internal structure of arguments.

The fundamental problem of learning a logical structure from examples in a probabilistic setting is not new: it is notably addressed in statistical relational learning (SRL) [9] and probabilistic inductive logic programming (PILP) [17]. However, argumentation plays no role in these approaches.

Some investigations focused on the relationship between inductive reasoning and non-monotonic reasoning akin to argument-based reasoning. [7], for example, gave an analysis of logical induction with a focus on *hypothesis generation* while [14] moved on with a characterisation for *hypothesis selection* with an illustration in rule-based argumentation. [10] investigated the induction of Defeasible Logic theories (possibly interpreted as an instantiated argumentation graph). In a parallel line of research, argumentation and inductive reasoning is combined in [13] to explain examples with expert's arguments. These works, however, do not cater for any probabilistic setting. Furthermore, they have little or no consideration, in the inductive process, for the different reasoning levels characterising argumentation frameworks (such as the labelling of arguments and the labelling of statements) along the fine granularity of possible labellings whereby, for example, an argument shall be labelled as justified, rejected, undecided or unexpressed.

Besides, case-based reasoning (CBR) [11] is often closely related to inductive reasoning since, from a set of cases, it shall form generalizations of these cases. When CBR shows dialectical features, as in legal reasoning, argumentation is a natural mean to model parties arguing about cases. For example, [16] proposed an early investigation of rule-based argumentation for reasoning with precedents represented by a set of attacking arguments, but the attacks over these arguments are given instead of being learnt.

The problem of learning attacks over arguments from arguments' statuses is different from the classic problem of learning a logical structure from statements, but our proposal may be used to position the labelling of arguments as a major reasoning step in structure learning, from the perspective of abstract argumentation and within a probabilistic framework.

As to the applications of the standalone problem, they regard any sequential setting where one or more agents repetitively argue. For each repetition, the status of every argument is justified, rejected, undecided, or unexpressed. The statuses of the arguments are observed, but the attack relations between arguments are unknown. The goal is to reconstruct the attacks.

This paper is organised as follows. Section 2 introduces the probabilistic abstract argumentation setting. Section 3 defines the problem we address. Then, we investigate the learning of argumentative structure in Section 4. with experimental insights in Section 5, before concluding.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

2. PROBABILISTIC ABSTRACT ARGUMENTATION

Our setting for probabilistic abstract argumentation is based on the proposal given in [19], which has the advantage of giving an explicit probability space and which fully relaxes probabilistic independences, even if there are no attack or sub-argument relations amongst arguments. The setting is built on abstract argumentation graphs [3].

DEFINITION 1 (ARGUMENTATION GRAPH). *An* argumentation graph *is a pair* $\langle \mathcal{A}, \rightsquigarrow \rangle$ *where* $\mathcal{A}$ *is a set of arguments, and* $\rightsquigarrow \subseteq \mathcal{A} \times \mathcal{A}$ *is a binary relation of attack.*

As for notation, given an argumentation graph $\mathcal{G} = \langle \mathcal{A}, \rightsquigarrow \rangle$, we write $\mathcal{A}_\mathcal{G} = \mathcal{A}$, and $\rightsquigarrow_\mathcal{G} = \rightsquigarrow$.

DEFINITION 2 (SUB-GRAPH). *A* sub-graph $\mathcal{H}$ *of an argumentation graph* $\mathcal{G} = \langle \mathcal{A}, \rightsquigarrow \rangle$ *is an argumentation graph* $\langle \mathcal{A}_\mathcal{H}, \mathcal{R}_\mathcal{H} \rangle$, *where* $\mathcal{A}_\mathcal{H} \subseteq \mathcal{A}$, *and* $\forall A, B \in \mathcal{A}_\mathcal{H}$, $(A \rightsquigarrow B) \in \rightsquigarrow_\mathcal{G}$ *iff* $(A \rightsquigarrow B) \in \rightsquigarrow_\mathcal{H}$.

Thus $\mathcal{H}$ is an induced sub-graph of $\mathcal{G}$ if it has exactly the attacks that appear in $\mathcal{G}$ over the same set of arguments.

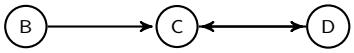


Figure 1: An argumentation graph. The argument B attacks the argument C, the arguments C and D attack each other.

Figure 2: The graph on the left is a sub-graph of the graph in Figure 1, while the graph on the right is not one of its sub-graphs.

Given an argumentation graph, the sets of arguments that are justified or rejected, that is, those arguments that shall survive or not to possible attacks, are computed using some semantics, and, for our purposes, we will consider Dung's grounded semantics [3] by labelling arguments as in [19], which is a slight adaptation of the labellings reviewed in [1] to fit a probabilistic setting. Accordingly, we will distinguish $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labellings and $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings. In a $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling, each argument is associated with one label which is either in, out, un, respectively meaning that the argument is justified, rejected, or undecided. 'in' means the argument is justified while a label 'out' indicates that it is rejected. The label 'un' marks the status of the argument as undecided. In a $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling, the label 'off' indicates that the argument is not expressed, that is, it does not occur.

DEFINITION 3 (LABELLING). *Let* $\mathcal{G}$ *denote an argumentation graph.*

- *A* $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling *of* $\mathcal{G}$ *is a total function* $L : \mathcal{A}_\mathcal{G} \to \{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$.
- *A* $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling *of* $\mathcal{G}$ *is a total function* $L : \mathcal{A}_\mathcal{G} \to \{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$.

In the remainder, we will write $\mathsf{in}(L)$ for $\{A | L(A) = \mathsf{in}\}$, $\mathsf{out}(L)$ for $\{A | L(A) = \mathsf{out}\}$, $\mathsf{un}(L)$ for $\{A | L(A) = \mathsf{un}\}$, and $\mathsf{off}(L)$ for $\{A | L(A) = \mathsf{off}\}$.

A $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling L will be represented as a tuple $\langle \mathsf{in}(L), \mathsf{out}(L), \mathsf{un}(L) \rangle$, and a $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling L as a tuple $\langle \mathsf{in}(L), \mathsf{out}(L), \mathsf{un}(L), \mathsf{off}(L) \rangle$.

Next the set of *complete labellings* is defined. As an argumentation graph may have several complete $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labellings, we will focus on the unique complete labelling with the smallest set of labels in, i.e. the grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling, see [1].

DEFINITION 4 (COMPLETE $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-LABELLING). *Let* $\mathcal{G}$ *denote an argumentation graph. A* complete $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling *of* $\mathcal{G}$ *is a* $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$*-labelling such that for every argument* A *in* $\mathcal{A}_\mathcal{G}$ *it holds that:*

- A *is labelled* in *iff all attackers of* A *are labelled* out,
- A *is labelled* out *iff* A *has an attacker labelled* in.

DEFINITION 5 (GROUNDED $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-LABELLING). *A complete labelling* L *is a* grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling *of an argumentation graph* $\mathcal{G}$ *if* $\mathsf{in}(L)$ *is minimal (w.r.t. set inclusion) amongst all complete* $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$*-labellings of* $\mathcal{G}$.

Note that, since a complete labelling is a total function, if an argument is not labelled in or out, then it is labelled un.

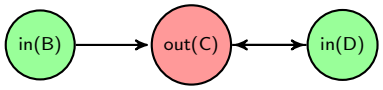


Figure 3: A grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling.

When some arguments are not expressed, we have *grounded* $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings, where only expressed arguments can effectively attack other expressed arguments.

DEFINITION 6 (GROUNDED $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-LABELLING). *Let* $\mathcal{G}$ *denote an argumentation graph and* $\mathcal{H}$ *an induced sub-graph of* $\mathcal{G}$. *A* grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling *of* $\mathcal{G}$ *is a* $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$*-labelling such that:*

- *every argument in* $\mathcal{A}_\mathcal{H}$ *is labelled according to the grounded* $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$*-labelling of* $\mathcal{H}$,
- *every argument in* $\mathcal{A}_\mathcal{G} \backslash \mathcal{A}_\mathcal{H}$ *is labelled* off.

An argumentation graph has a unique grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling, but it has as many grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings as sub-graphs.

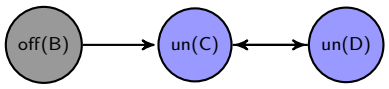


Figure 4: A grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling.

As for notational matters, a complete $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling will be abbreviated as $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}^c$-labelling, and a grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling as $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}^g$-labelling. By doing so, we can denote the set of $\mathcal{S}$-labellings of an argumentation graph $\mathcal{G}$ as $\mathcal{L}^\mathcal{S}_\mathcal{G}$, and each set will basically constitute a possible sample

space (i.e. the set of possible outcomes) of our probabilistic setting for argumentation.

As an intuitive account of the probabilistic setting, an agent is in front of a bag of $\mathcal{S}$-labellings, and this agent observes a labelling L grasped from the bag with a probability $P(\{L\})$. In other words, a labelling of an argumentation graph is an outcome, and each outcome is associated with a probability. This view on probabilistic argumentation is formally captured by the following definition of probabilistic argumentation frames.

Definition 7 (Prob. argumentation frame). *A probabilistic argumentation frame is a tuple $(\mathcal{G}, \mathcal{S}, (\Omega, F, P))$ where $\mathcal{G}$ is an argumentation graph (called the hypothetical argumentation frame), $\mathcal{S}$ is the labelling specification and, (Ω, F, P) is a probability space such that:*

- *the sample space Ω is the set of $\mathcal{S}$-labellings of the hypothetical argumentation frame $\mathcal{G}$, $\Omega = \mathcal{L}^{\mathcal{S}}_{\mathcal{G}}$,*
- *the σ-algebra F is the power set of Ω,*
- *the probability function P from $F(\Omega)$ to $[0, 1]$ satisfies Kolmorogov axioms.*

As an illustration, assume a hypothetical argumentation frame as drawn in Figure 1, the sample space of grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings is shown in the table below.

B	in	in	in	in	off	off	off	off
C	out	out	off	off	un	in	off	off
D	in	off	in	off	un	off	in	off

Table 1: Sample space as the set of grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings of the argumentation graph drawn in Figure 1.

In the remainder of this paper, we will focus on the grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling, thus $\mathcal{S}$ will hold for the specification of grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings. We will consider a sequence of grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings drawn from a distribution, and our goal is to find an argumentation graph that accounts for these labellings.

3. PROBLEM SETTING

From now, we assume an empirical probability distribution $P_{\mathcal{O}}$ over a set $\mathcal{O}$ of observed grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings, such that these observed labellings are drawn from a probability distribution of a probabilistic argumentation frame $\langle \mathcal{G}, \mathcal{S}, (\Omega, F, P)\rangle$. The distribution $P_{\mathcal{O}}$ is thus the empirical approximation of the distribution P. The hypothetical argumentation frame $\mathcal{G}$ is called the source argumentation graph, since it is going to be the 'source' of a stream of labellings, and P is called the source distribution.

We readily remark that different argumentation graphs may be indistinguishable in the sense that they have, or can account for, the same observed labellings. For example, the graphs $\mathcal{G}_1 = \langle\{\mathsf{A}, \mathsf{B}\}, \{\mathsf{A} \rightsquigarrow \mathsf{A}, \mathsf{A} \rightsquigarrow \mathsf{B}\}\rangle$ and $\mathcal{G}_2 = \langle\{\mathsf{A}, \mathsf{B}\}, \{\mathsf{A} \rightsquigarrow \mathsf{A}, \mathsf{A} \rightsquigarrow \mathsf{B}, \mathsf{B} \rightsquigarrow \mathsf{A}\}\rangle$ have the same grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings. As the graph $\mathcal{G}_1$ has less attacks than the graph $\mathcal{G}_2$, then one might consider that the $\mathcal{G}_1$ is a better graph to account for some observed labellings. However it may the case that the source underlying graph is in fact $\mathcal{G}_2$. This example shows that, in some cases, we cannot aim to induce the exact source graph $\mathcal{G}$ of a probabilistic argumentation frame $\langle \mathcal{G}, \mathcal{S}, (\Omega, F, P)\rangle$ underlying a distribution of observed labellings, but this does not prohibit us to find an argumentation graph $\mathcal{X}$ that shall explain (to different degrees) the observed labellings.

To measure how much an argumentation graph $\mathcal{X}$ explains a set $\mathcal{O}$ of observed labellings along its empirical distribution $P_{\mathcal{O}}$, we will assume an explanatory measure of this graph, denoted $M(\mathcal{X}, P_{\mathcal{O}})$, such that the value of this measure is maximal when the graph $\mathcal{X}$ is the source argumentation graph $\mathcal{G}$.

Many different explanatory measures can be considered. Here, an example of a measure that we shall call the *expected explanatory utility* of a graph $\mathcal{X}$ w.r.t. a distribution $P_{\mathcal{O}}$ of a set $\mathcal{O}$ of observed labellings:

$$M(\mathcal{X}, P_{\mathcal{O}}) = \sum_{L \in \mathcal{O}} P_{\mathcal{O}}(L) \cdot \mu(\mathcal{X}, L) \tag{1}$$

where $\mu(\mathcal{X}, L)$ is the explanatory utility of the graph $\mathcal{X}$ to explain the labelling L. We may consider that the graph $\mathcal{X}$ is useful to explain a labelling L when the similarity of the labelling L and the corresponding grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling of $\mathcal{X}$, denoted L' (the labelling with the same arguments labelled off) is maximised. Accordingly, we may measure the explanatory utility with the following explanatory index with a Jaccard flavour:

$$\mu(\mathcal{X}, L) = \frac{s}{s_{\mathsf{in}} + s_{\mathsf{out}} + s_{\mathsf{un}} + s} \tag{2}$$

where[1]

$$s = |\mathsf{in}(L) \cap \mathsf{in}(L')| + |\mathsf{out}(L) \cap \mathsf{out}(L')| + |\mathsf{un}(L) \cap \mathsf{un}(L')| \tag{3}$$

$$s_{\mathsf{in}} = |\mathsf{in}(L) \cap \mathsf{out}(L')| + |\mathsf{in}(L) \cap \mathsf{un}(L')| \tag{4}$$

$$s_{\mathsf{out}} = |\mathsf{out}(L) \cap \mathsf{in}(L')| + |\mathsf{out}(L) \cap \mathsf{un}(L')| \tag{5}$$

$$s_{\mathsf{un}} = |\mathsf{un}(L) \cap \mathsf{in}(L')| + |\mathsf{un}(L) \cap \mathsf{out}(L')| \tag{6}$$

Hence $\mu(\mathcal{X}, L) \in [0, 1]$: if all the labels mismatch then $\mu(\mathcal{X}, L) = 0$, if all the labels match then $\mu(\mathcal{X}, L) = 1$. Consequently, if all the labellings totally mismatch then $M(\mathcal{X}, P_{\mathcal{O}}) = 0$, if all the labellings totally match then $M(\mathcal{X}, P_{\mathcal{O}}) = \sum_{L \in \mathcal{O}} P_{\mathcal{O}}(L)$, and thus $M(\mathcal{X}, P_{\mathcal{O}}) = 1$.

We reckon that the choice of an explanatory measure shall much depend on the envisaged application, and thus, we do not further discuss this point. However, we are now prepared to precise our abstract structure learning problem.

Given:

- an empirical probability distribution $P_{\mathcal{O}}$ of labellings over a set $\mathcal{O}$ of observed labellings (of a set $\mathcal{A}$ of observed arguments), such that these observed labellings are drawn from a probability distribution of a probabilistic argumentation frame $\langle \mathcal{G}, \mathcal{S}, (\Omega, F, P)\rangle$.
- a set $\mathfrak{H}$ of hypothetical argumentation graphs $\langle \mathcal{A}, \rightsquigarrow\rangle$.

find (or induce or learn)

- a hypothetical argumentation graph $\mathcal{X} \in \mathfrak{H}$ such that the explanatory measure $M(\mathcal{X}, P_{\mathcal{O}})$ is maximised.

[1]Here, s is such that we ignore the labelling where all the arguments are labelled off.

If we find an argumentation graph indistinguishable from the source argumentation graph $\mathcal{G}$ underlying the observed labellings, then the explanatory measure will be maximised. Even if the source argumentation graph $\mathcal{G}$ is not found, we may confirm or discard some attack relations, which shall nevertheless appear valuable to inform (i.e. explain) the observed labellings.

4. LEARNING ATTACKS

Our problem assumes a set $\mathfrak{H}$ of hypothetical argumentation graphs, and we have to find in it an argumentation graph to account for some labellings observed in sequence. To build this set, we start from the complete argumentation graph induced by the set of observed arguments, i.e. a graph where all the arguments attack each other.

DEFINITION 8 (COMPLETE ARGUMENTATION GRAPH). *A* complete argumentation graph *induced by a set of arguments $\mathcal{A}$ is an argumentation graph $\langle \mathcal{A}, \rightsquigarrow \rangle$, where* $\rightsquigarrow = \{A \rightsquigarrow B | A, B \in \mathcal{A}\}$.

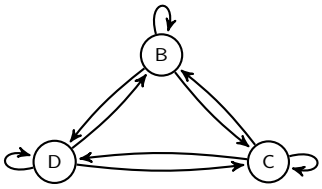


Figure 5: A complete argumentation graph induced by the arguments $\{\mathsf{B}, \mathsf{C}, \mathsf{D}\}$.

Given a probabilistic argumentation graph $\langle \mathcal{G}, \mathcal{S}, (\Omega, F, P) \rangle$, the hypothetical argumentation frame $\mathcal{G}$ is necessarily an 'attack sub-graph' (defined below) of the complete argumentation graph $\mathcal{C}$ induced by the set of arguments $\mathcal{A}_\mathcal{G}$.

DEFINITION 9 (ATTACK SUB-GRAPH). *An* attack sub-graph *of an argumentation graph $\langle \mathcal{A}, \rightsquigarrow \rangle$ is an argumentation graph $\langle \mathcal{A}, \rightsquigarrow' \rangle$ such that $\rightsquigarrow' \subseteq \rightsquigarrow$.*

For example, the graph in Figure 1 is an attack sub-graph of the complete graph shown in Figure 5.

A brute force approach for our problem is to start from the complete argumentation graph $\mathcal{C}$ induced by the set of observable arguments, and then consider every attack sub-graphs of $\mathcal{C}$ till we find a graph $\mathcal{X}$ covering all the observed labellings. However this approach is of course not efficient.

As the brute force approach is not efficient, we investigate here an alternative: we start from a complete hypothetical argumentation graph induced from a set of arguments, then, based on the labellings observed in sequence, we will attach a *credit value* to any hypothetical attack relation, and we shall discard or confirm attacks based on their credits. Since we attach a credit value (or let us say a weight) to any hypothetical attack, we resort to *weighted argumentation graphs* (see [5]):

DEFINITION 10 (WEIGHTED ARGUMENTATION GRAPH). *A* weighted argumentation graph *is a triple $\langle \mathcal{A}, \rightsquigarrow, v \rangle$ such that $\langle \mathcal{A}, \rightsquigarrow \rangle$ is an argumentation graph, and $v : \rightsquigarrow \rightarrow \mathbb{R}$ is a function assigning a real valued weight to every attack.*

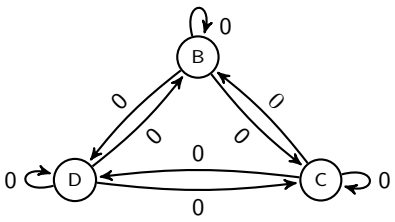


Figure 6: A weighted argumentation graph, where all the weights have a value 0.

Note that weighted argumentation graphs in [5] are used for different investigations: [5] uses weights to introduce an inconsistency budget, whereas we are simply using weighted argumentation graphs to store a number (that we call a credit) to every attack, with the idea that this numerical approach shall ease the treatment of noisy settings in future investigations.

In the remainder, given a weighted argumentation graph $\mathcal{C}$, we will say that an attack $A \rightsquigarrow B$ is

- discarded if $v_{A \rightsquigarrow B} < 0$,
- confirmed if $v_{A \rightsquigarrow B} > 0$,
- undecided if $v_{A \rightsquigarrow B} = 0$,
- possible if $v_{A \rightsquigarrow B} \geq 0$ (in this case the argument A is called a possible attacker of B).

When we abstract from the weight function of a weighted argumentation graph $\langle \mathcal{A}, \rightsquigarrow, v \rangle$, then we shall straightforwardly consider the corresponding argumentation graph $\langle \mathcal{A}, \rightsquigarrow \rangle$ where the weight function is omitted.

To learn attacks, we start from the weighted complete argumentation graph $\mathcal{C}$ where an initial credit is attached to every attack of $\mathcal{C}$, and in order to cater for the observed labellings, we shall change the value of the credits to discard or confirm attacks. To change the credits, we consider simple theorems considering labellings of single arguments or pairs of arguments to induce the presence of some attacks.

Theorems. Let $\mathcal{G}$ be an argumentation graph, L be a grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling of $\mathcal{G}$, and A, B and C denote (not necessarily distinct) arguments in $\mathcal{A}_\mathcal{G}$.

THEOREM 1. *If A and B are labelled* in *(i.e. $L(A) = \mathsf{in}$ and $L(B) = \mathsf{in}$), then there is no attack $B \rightsquigarrow A$ (i.e. $B \rightsquigarrow A \notin \rightsquigarrow_\mathcal{G}$).*

PROOF. By contradiction. Suppose there is an attack $B \rightsquigarrow A$. Since B is in and L is complete, we conclude that A is out, which contradicts our assumption that A is in. □

THEOREM 2. *If A is labelled* in *and B is labelled* un*, then there are no attacks $A \rightsquigarrow B$ and $B \rightsquigarrow A$.*

PROOF. By contradiction. (i) Suppose there is an attack $A \rightsquigarrow B$. Since A is in and L is complete, B is out, which contradicts our assumption that B is un. (ii) Suppose that there is an attack $B \rightsquigarrow A$. Since B is un and L is complete, A is not in, which contradicts our assumption that A is in. □

THEOREM 3. *If A is labelled* un*, and any possible attacker C of A is labelled* out *or* off *($C \neq A$), then there is an attack $A \rightsquigarrow A$.*

PROOF. By contradiction. Suppose that there is no attack $A \rightsquigarrow A$. Since L is complete, we conclude that A is labelled in, which contradicts our assumption that A is un. □

THEOREM 4. *If A is labelled un and B is labelled un, and there are no attacks $A \rightsquigarrow A$ and $B \rightsquigarrow B$, and any possible attacker C of A or B is labelled out or off ($C \neq A$ and $C \neq B$), then there is an attack $A \rightsquigarrow B$ and $B \rightsquigarrow A$.*

PROOF. By contradiction. Suppose that there are no attacks $A \rightsquigarrow B$ and $B \rightsquigarrow A$. Since L is grounded, we conclude that A and B are not labelled un, which contradicts our assumption that A and B are un. □

THEOREM 5. *If A is out and B is in, and any possible attacker C of A is labelled out or un or off ($C \neq A$ and $C \neq B$), then there is an attack $B \rightsquigarrow A$ and no attack $A \rightsquigarrow B$.*

PROOF. By contradiction. Suppose it is not the case that "there is an attack $B \rightsquigarrow A$ and there is no attack $B \rightsquigarrow A$". Since L is grounded, A is not out and B is not in, which contradicts our assumption that A is out and B is in. □

Although these theorems are simple, they are at the core of our approach. They are the basis of the credit rules used in Algorithm 1 to progressively confirm or discard attacks, by increasing or decreasing their credits each time a labelling is observed.

DEFINITION 11 (CREDIT RULES). *Let $\mathcal{C} = \langle \mathcal{A}, \rightsquigarrow, v \rangle$ denote a weighted complete argumentation graph, and L any grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling of $\mathcal{C}$. For any argument A and B labelled in L, we have the following rules updating the credits of attacks:*

$\mathsf{R}_1(L, \mathcal{C})$: *If $L(A) = \mathsf{in}$ and $L(B) = \mathsf{in}$, then*

$$v_{B \rightsquigarrow A} \leftarrow v_{B \rightsquigarrow A} - 1$$

$\mathsf{R}_2(L, \mathcal{C})$: *If $L(A) = \mathsf{in}$ and $L(B) = \mathsf{un}$, then*

$$v_{B \rightsquigarrow A} \leftarrow v_{B \rightsquigarrow A} - 1; \quad v_{A \rightsquigarrow B} \leftarrow v_{A \rightsquigarrow B} - 1$$

$\mathsf{R}_3(L, \mathcal{C})$: *If $L(A) = \mathsf{un}$, $L(B) = \mathsf{un}$, and for any possible attacker C of A or B ($C \neq A$ and $C \neq B$), $L(C) = \mathsf{out}$ or $L(C) = \mathsf{off}$, then*

$$v_{B \rightsquigarrow A} \leftarrow v_{B \rightsquigarrow A} + 1; \quad v_{A \rightsquigarrow B} \leftarrow v_{A \rightsquigarrow B} + 1$$

$\mathsf{R}_4(L, \mathcal{C})$: *If $L(A) = \mathsf{out}$, $L(B) = \mathsf{in}$, and for any possible attacker C of A ($C \neq A$ and $C \neq B$) we have $L(C) = \mathsf{out}$ or $L(C) = \mathsf{un}$ or $L(C) = \mathsf{off}$, then*

$$v_{B \rightsquigarrow A} \leftarrow v_{B \rightsquigarrow A} + 1; \quad v_{A \rightsquigarrow B} \leftarrow v_{A \rightsquigarrow B} - 1$$

On the basis of the credit rules, we propose an algorithm (Algorithm 1) to learn the attack relations: it iteratively credits the attack relations until some predefined computational budget (typically time or iteration constraint) is reached or until every attack is either confirmed or discarded, at which point the loop is halted and the argumentation graph induced so far is returned.

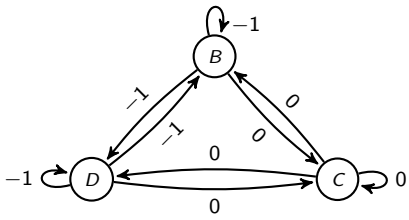
The algorithm starts from a weighted argumentation graph $\mathcal{C}$, where the credit of every attack is initialised (line 2) - if we are ignorant about the status of an attack, then its credit is initialised to 0. Every time that a labelling is observed (line 4), credits are possibly updated (line 5), until all attacks are confirmed or discarded, or some computational budget is reached. Notice that discarded attacks can be dumped (line 6) to accelerate the application of credit rules. When the loop ends, the weighted argumentation graph $\mathcal{C}$ is pruned (line 8) by dumping any undecided or discarded attacks, to return the argumentation graph with only the confirmed attacks.

Algorithm 1 Attack learning algorithm.

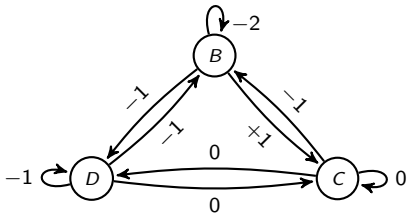
1: **input** The weighted complete argumentation graph $\mathcal{C} = \langle \mathcal{A}_\mathcal{C}, \rightsquigarrow_\mathcal{C}, v \rangle$.
2: For any arguments $A, B \in \mathcal{A}_\mathcal{C}$, initialise credits $v_{A \rightsquigarrow B}$.
3: **while** there is an attack which is neither confirmed nor discarded, or within computational budget **do**
4: Get a labelling instance L.
5: Credit the attack relations of $\mathcal{C}$ using the credit rules $\mathsf{R}_1(L, \mathcal{C})$, $\mathsf{R}_2(L, \mathcal{C})$, $\mathsf{R}_3(L, \mathcal{C})$, $\mathsf{R}_4(L, \mathcal{C})$.
6: $\mathcal{C} \leftarrow pruneDiscardedAttacks(\mathcal{C})$. (optional)
7: **end while**
8: $\mathcal{X} \leftarrow \text{prune}(\mathcal{C})$.
9: **return** the argumentation graph $\mathcal{X}$.

EXAMPLE 1. *Let us illustrate Algorithm 1. Consider the argumentation graph shown in Figure 1 as our source graph of a stream of labellings that we will observe. All attacks are initialised with a credit 0, and thus we start from the weighted argumentation graph as given in Figure 6.*

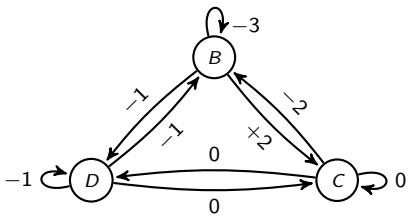
1. *Suppose that the 1st observed labelling instance is $(\{\mathsf{B}, \mathsf{D}\}, \{\mathsf{C}\}, \emptyset, \emptyset)$. Using the credit rule R_1, the weights of the weighted argumentation graph are updated as follows.*



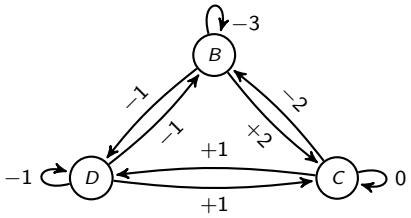
2. *Suppose the 2nd observed labelling instance is $(\{\mathsf{B}\}, \{\mathsf{C}\}, \emptyset, \{\mathsf{D}\})$. Using credit rules R_1 and R_4, the weighted argumentation graph is updated as follows.*



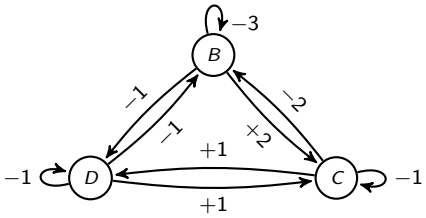
3. *Suppose the 3rd observed labelling instance is again $(\{\mathsf{B}\}, \{\mathsf{C}\}, \emptyset, \{\mathsf{D}\})$. We use the credit rules R_1 and R_4.*



4. Suppose the 4th observed labelling is $(\emptyset, \emptyset, \{\mathsf{C}, \mathsf{D}\}, \{\mathsf{B}\})$. *We use the credit rule* R_3.



5. Suppose the 5th labelling instance is $(\{\mathsf{C}\}, \emptyset, \emptyset, \{\mathsf{B}, \mathsf{D}\})$. *We use the credit rule* R_1 *for the update.*



At this stage, all attacks are either confirmed or discarded, and thus we can prune the weighted argumentation graph by dumping all discarded attacks. Finally we return the argumentation graph with only the confirmed attacks, which is the argumentation graph shown in Figure 1.

As the example illustrates, the algorithm works 'on the fly' and thus a major benefit is that we do not have to keep in memory the observed labellings. However we have to keep in memory all the credits of all possible attacks between any pair of arguments amongst $|\mathcal{A}_\mathcal{G}|$ arguments of a complete argumentation graph $\mathcal{G}$. The space complexity is thus quadratic.

THEOREM 6 (SPACE COMPLEXITY). *The space complexity of Algorithm 1 is quadratic.*

PROOF. Trivial, since the number of possible attacks amongst $n = |\mathcal{A}_\mathcal{G}|$ arguments of a complete argumentation graph $\mathcal{G}$ is n^2. □

Concerning time complexity, the burden stems from the application of credit rules to all possible attacks for any pair of arguments amongst $|\mathcal{A}_\mathcal{G}|$ arguments of a complete argumentation graph $\mathcal{G}$.

THEOREM 7 (TIME COMPLEXITY). *The time complexity of applying credit rules is cubic.*

PROOF. In the worst case, the most demanding rules R_3 and R_4 have to check the status of $n = |\mathcal{A}_\mathcal{G}|$ attackers for the arguments of any n^2 possible attacks amongst arguments. □

In practice, the computational complexity is often drastically decreased by pruning discarded attacks of the argumentation graph $\mathcal{C}$ as optionally proposed in line 6 of the algorithm.

In case of active learning (i.e. the algorithm can decide which labellings to consider in line 4 of Algorithm 1), key labellings (see Definition 13) can be observed to straightforwardly evaluate hypothetical attacks. A *key labelling* is a $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling where a set of arguments is 'isolated', i.e. all the arguments not in this set are labelled off.

DEFINITION 12. *A* n-isolated grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling *is a grounded* $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$*-labelling with exactly* n *distinct arguments not labelled* off.

DEFINITION 13 (KEY LABELLING). *A* key labelling *is a* 1*-isolated grounded* $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$*-labelling or a* 2*-isolated grounded* $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$*-labelling.*

For instance, any labelling in Table 1 is a key labelling, except the labellings $(\{\mathsf{B}, \mathsf{D}\}, \{\mathsf{C}\}, \emptyset, \emptyset)$ and $(\emptyset, \emptyset, \emptyset, \{\mathsf{B}, \mathsf{C}, \mathsf{D}\})$.

Table 2 summarises the coverage of credit rules w.r.t. every possible key grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling of two arguments A and B whose the attack relation is unknown.

Table 2: Coverage of key labellings by the credit rules.

A	in	in	in	in	un	un
B	in	out	un	off	un	off
	R_1	R_4	R_2	R_1	R_3	R_3

Every key labelling is covered. However, as illustrated in Section 3, we cannot distinguish the attacks between two arguments from a labelling where both arguments are labelled un and at least one argument self-attacks. We have thus to resolve to induce argumentation graphs indistinguishable from source graphs.

THEOREM 8. *Let* $\mathcal{G}$ *denote a source argumentation graph, let* $n = |\mathcal{A}_\mathcal{G}|$ *denote the number of arguments in* $\mathcal{G}$*. An argumentation graph, indistinguishable from the source graph* $\mathcal{G}$*, is induced by observing its* $n \cdot (n+1)/2$ *key labellings.*

PROOF. The number of 1-isolated grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings of a graph with n arguments is n, while the number of 2-isolated grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings is $C(2, n) = n \cdot (n-1)/2$[2]. Thus the number of key labellings is $n \cdot (n+1)/2$. By observing all the $n \cdot (n+1)/2$ key labellings, we can thus confirm or discard all the attack relations amongst the n arguments (see Table 2). Finally, the confirmed attacks are the attacks of a graph indistinguishable from the source graph. □

In practice, Theorem 8 is interesting because it implies that, in an active learning setting where it is possible to choose to observe some labellings in particular, one shall choose to observe $n \cdot (n+1)/2$ key labellings to straightforwardly induce an argumentation graph indistinguishable from the source graph (in this case the explanatory measure is maximised). However, the number of labellings observed to induce the

[2]using standard notation, $C(k, n) = \frac{n!}{k!(n-k)!}$, $k \leq n$.

source graph can be inferior to the number of key labellings, because a labelling may contain multiple isolated (pairs of) arguments.

If active learning is not possible, and if the algorithm terminates before any attack is confirmed or discarded, then the returned argumentation graph $\mathcal{X}$ may be such that there exists some labellings of $\mathcal{X}$ which are not in the set of observed labellings, and there exists some observed labellings which are not labellings of $\mathcal{X}$. So, the algorithm may never confirm or discard all the attacks, and for this reason, we will assume in the remainder that the underlying probabilistic argumentation frame $\langle \mathcal{G}, \mathcal{S}, (\Omega, F, P) \rangle$ of the observed labelling is such that for any labelling L in the sample space Ω, we have $P(\{L\}) > 0$. With this assumption, the algorithm will always empty the set of undecided attacks.

THEOREM 9 (TERMINATION). *If the computational budget is finite, then Algorithm 1 terminates, else it terminates almost surely (i.e. it terminates with probability one).*

PROOF. If the computational budget is finite, then Algorithm 1 terminates when the computational budget is reached, else the probability that any key labelling get drawn is one (since $P(\{L\}) > 0$). From the key labellings, the attack relations between any pair of arguments are confirmed or discarded, which terminates the loop while. □

THEOREM 10 (SOUNDNESS AT TERMINATION). *If the computational budget is infinite, Algorithm 1 returns almost surely an argumentation graph indistinguishable from the source argumentation graph.*

PROOF. From the key labellings, we induce an argumentation graph indistinguishable from the source argumentation graph. Since all these labellings appear almost surely, then the algorithm returns almost surely an argumentation graph indistinguishable from the source argumentation graph. □

In this regard, the waiting time to collect the set of key labellings is the waiting time akin to a partial collection in the coupon collector problem, see e.g. [8].

If active learning is not possible and the computational budget is finite, some attacks may remain undecided, and thus we may be willing to empty the set of undecided attacks, so that the graph $\mathcal{X}$ can give full explanation of the observed labellings. To empty the set of undecided attacks, a solution consists in approaching the undecided attacks with a brute force, or any other more subtle technique which we do not address here. In the brute force approach, and using Occam's razor, the graph with the fewest attacks shall be selected, but we leave such considerations for future work.

5. EXPERIMENTS

To obtain experimental insights, we created 100 artificial (and thus domain independent) probabilistic argumentation frames $\langle \mathsf{G}, \mathcal{S}, (\Omega, F, P_i) \rangle$ $(i \in \{1, \ldots, 100\})$, all based on the source argumentation graph G shown in Figure 7. This graph has no self-attacking arguments, and thus we can aim to induce it exactly. Since we have 12 arguments, then a brute approach for finding the graph G would have to consider 2^{144} hypothetical graphs.

Each probabilistic argumentation frame had a different entropy. The entropy of the distribution P_i of a probabilistic

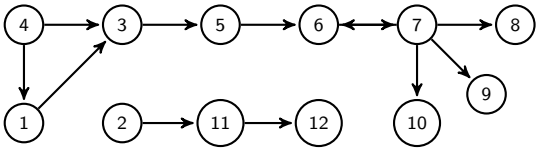


Figure 7: Source argumentation graph.

argumentation frame $\langle \mathsf{G}, \mathcal{S}, (\Omega, F, P_i) \rangle$ is defined as follows:

$$entropy(P_i) = -\sum_{L \in \Omega} P_i(L) \cdot \ln(P_i(L)) \qquad (7)$$

To ensure meaningful experiments, we set a probability 0 for any labelling where all arguments were labelled off.

For every distribution, we run the Algorithm 1 on a stream of labelling instances (drawn from the considered distribution) in order to induce a graph maximising the explanation measure.

As we observed a sequence of n labelling instances, some instances may not be distinct since a labelling may be drawn twice or more. Accordingly, we have to distinguish the number n of labelling instances, and the number m of distinct labelling instances which are observed ($m \leq n$). In Example 1, we observed 6 labelling instances, but we observed 5 distinct labelling instances because the labelling instances at the steps 2 and 3 are the same.

We stopped the algorithm as soon as:

- all the attacks of the source graph were included in the set of confirmed attacks, and the number of undecided attacks was less or equal to 10 (simulating so a brute force search on the remaining undecided attacks), or
- $n = 10^6$ labelling instances were observed.

The number of observed labellings at termination is shown in Figure 8 and the associated expected explanatory utility (as formulated in Section 3) of the induced argumentation graph is given in Figure 9.

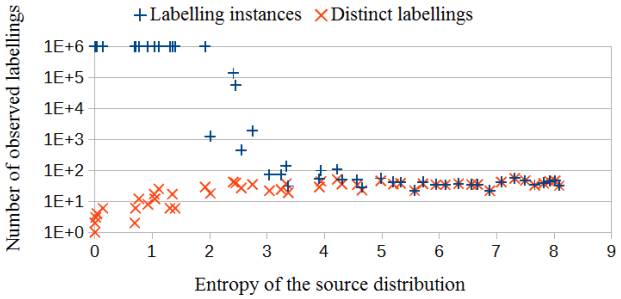


Figure 8: Number of labelling instances and distinct labellings observed at termination.

As expected, the lower the entropy, the longer it took to induce the source argumentation graph. Whatever the entropy, a few distinct labelling instances (less than one hundred) were necessary to exactly induce the source graph - to compare to the 2^{144} possible attack sub-graphs. For any entropy above 2, the source argumentation graph G was exactly induced. Furthermore, Figure 9 shows that even the

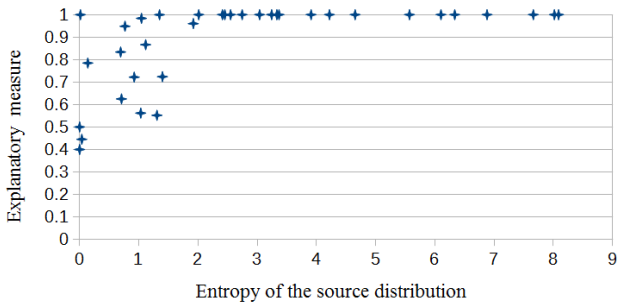


Figure 9: Expected explanatory utility of the induced argumentation graph at termination w.r.t. the observed labellings.

original graph is not retrieved, the expected explanatory utility of the returned graph may be close to the maximum.

In a second set of experiments, we evaluated the approach on different distributions, all with an entropy 3 ± 0.05, based on source graphs with different number of arguments, and with up to 100 arguments (i.e. up to 2^{10000} hypothetical graphs).

The number of labelling instances and distinct labelling instances observed at termination is shown in Figure 10 for each distribution.

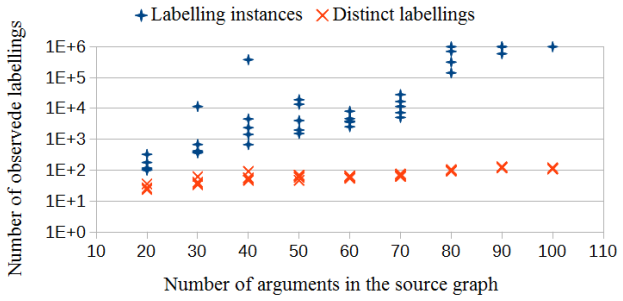


Figure 10: Number of labellings at termination.

The more the number of observed arguments, the longer the time to guess the source graph. Even when the source graph was not found at termination, the expected explanatory utility of the induced graph was above 0.99 at termination (this pretty flat figure is not shown).

Since the source argumentation graphs were exactly induced in most cases (as expected from Theorem 10) and the expected explanatory utility of the induced graphs was above 0.99 at termination in even more cases, the experiments suggest that induced graphs 'make sense' in general. The principal reason holds in that if an attack relation is discarded then this attack is indeed not part of the source graph, and when there are no self-attacking arguments, if an attack relation is confirmed then this attack is indeed part of the source graph, as ensured by backing our approach with Theorems 1-5.

6. CONCLUSION

We settled a problem of abstract structure learning regarding the induction of attacks between arguments in a probabilistic framework for abstract argumentation: labellings of arguments are observed and the goal is to learn attacks between arguments to account for the observed labellings by maximising an explanation measure.

To address this problem of abstract structure learning, we proposed to use five simple theorems about the attacks which can be learned from a labelling of arguments. Based on these theorems, we proposed an anytime and 'on the fly' algorithm taking as input a sequence of labellings, weighting the credit of attacks using credit rules, and returning an argumentation graph meant to account for the observed labellings.

We showed that if the computational budget is infinite, then the algorithm returns - almost surely - an argumentation graph indistinguishable from the source graph. In case of active learning, such an argumentation graph (of n arguments) is returned for sure (and thus the explanatory measure is maximised), by observing its $n \cdot (n+1)/2$ key labellings. In practice, it is possible to induce some source graphs with less labellings than the number of key labellings. The experiments showed that the time to induce a graph indistinguishable from source graph is largely dependent on the entropy of the source distribution of labellings. The higher the entropy, the faster the induction is performed.

As future developments, one may regard different credit rules and other types of labellings, possibly in the light of insights regarding their properties (see e.g. [18, 4]). One may also regard settings with structured arguments and more sophisticated updates of the credits to deal with some types of noise. As the proposed algorithm works with streams of labellings, it may be coupled with a so-called argumentative Boltzmann machine [19, 20] (by showing the labelling grasped line 4 to the machine). Doing so, the induction of the argumentation graph shall occur at the same time as the learning of the probability distribution of labellings, giving thus explanatory ability to the neural network.

Acknowledgements

We would like to thank the helpful AAMAS reviewers. NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre for Excellence Program. Part of the work was supported by the Marie Curie Intra-European Fellowship PIEF-GA-2012-331472.

REFERENCES

[1] P. Baroni, M. Caminada, and M. Giacomin. An introduction to argumentation semantics. *Knowledge Engineering Review*, 26(4):365–410, 2011.

[2] P. Dondio. Toward a computational analysis of probabilistic argumentation frameworks. *Cybernetics and Systems*, 45(3):254–278, 2014.

[3] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

[4] P. Dunne, W. Dvorák, T. Linsbichler, and S. Woltran. Characteristics of multiple viewpoints in abstract argumentation. *Artif. Intell.*, 228:153–178, 2015.

[5] P. Dunne, A. Hunter, P. McBurney, S. Parsons, and M. Wooldridge. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artif. Intell.*, 175(2):457–486, 2011.

[6] B. Fazzinga, S. Flesca, and F. Parisi. On the complexity of probabilistic abstract argumentation frameworks. *ACM Trans. Comput. Logic*, 16(3):22:1–22:39, June 2015.

[7] P. A. Flach. Rationality postulates for induction. In *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 267–281. Morgan Kaufmann, 1996.

[8] P. Flajolet, D. Gardy, and L. Thimonier. Birthday paradox, coupon collectors, caching algorithms and self-organizing search. *Discrete Appl. Math.*, 39(3):207–229, Nov. 1992.

[9] L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007.

[10] B. Johnston and G. Governatori. Induction of defeasible logic theories in the legal domain. In *Proceedings of the 9th International Conference on Artificial Intelligence and Law*, pages 204–213. ACM, 2003.

[11] J. Kolodner. *Case-based Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[12] H. Li, N. Oren, and T. J. Norman. Probabilistic argumentation frameworks. In *Revised Selected Papers of the 1st International Workshop on Theory and Applications of Formal Argumentation.*, Lecture Notes in Computer Science, pages 1–16. Springer, 2011.

[13] M. Mozina, J. Zabkar, and I. Bratko. Argument based machine learning. *Artif. Intell.*, 171(10-15):922–937, 2007.

[14] S. Ontañón, P. Dellunde, L. Godo, and E. Plaza. A defeasible reasoning model of inductive concept learning from examples and communication. *Artif. Intell.*, 193:129–148, Dec. 2012.

[15] S. Polberg and D. Doder. Probabilistic abstract dialectical frameworks. In *Proceedings of the 14th European Conference on Logics in Artificial Intelligence*, pages 591–599. Springer, 2014.

[16] H. Prakken and G. Sartor. Reasoning with precedents in a dialogue game. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law*, ICAIL '97, pages 1–9, New York, NY, USA, 1997. ACM.

[17] L. D. Raedt, P. Frasconi, K. Kersting, and S. Muggleton, editors. *Probabilistic Inductive Logic Programming - Theory and Applications*. Springer Berlin Heidelberg, 2008.

[18] T. Rienstra, C. Sakama, and L. W. N. van der Torre. Persistence and monotony properties of argumentation semantics. In *Revised Selected Papers of the 3rd International Workshop on Theory and Applications of Formal Argumentation*, pages 211–225. Springer, 2015.

[19] R. Riveret, D. Korkinof, M. Draief, and J. V. Pitt. Probabilistic abstract argumentation: an investigation with boltzmann machines. *Argument & Computation*, 6(2):178–218, 2015.

[20] R. Riveret, J. Pitt, D. Korkinof, and M. Draief. Neuro-symbolic agents: Boltzmann machine and Probabilistic Abstract Argumentation with Sub-arguments. In *Proceedings of the 14th International Joint Conference on Autonomous Agents & Multiagent Systems*. IFAAMAS, 2015.

[21] M. Thimm. A probabilistic semantics for abstract argumentation. In *Proceedings of the 20th European Conference on Artificial Intelligence*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, pages 750–755. IOS Press, 2012.