Neural-Symbolic Probabilistic Argumentation Machines*

Régis Riveret[1*], **Son Tran**[2†], **Artur d'Avila Garcez**[3]
[1]Commonwealth Scientific and Industrial Research Organisation, Australia
[2]University of Tasmania, Australia
[3]City University London, United Kingdom
regis.riveret@csiro.au, sn.tran@utas.edu.au, a.garcez@city.ac.uk

Abstract

Neural-symbolic systems combine the strengths of neural networks and symbolic formalisms. In this paper, we introduce a neural-symbolic system which combines restricted Boltzmann machines and probabilistic semi-abstract argumentation. We propose to train networks on argument labellings explaining the data, so that any sampled data outcome is associated with an argument labelling. Argument labellings are integrated as constraints within restricted Boltzmann machines, so that the neural networks are used to learn probabilistic dependencies amongst argument labels. Given a dataset and an argumentation graph as prior knowledge, for every example/case K in the dataset, we use a so-called K-maxconsistent labelling of the graph, and an explanation of case K refers to a K-maxconsistent labelling of the given argumentation graph. The abilities of the proposed system to predict correct labellings were evaluated and compared with standard machine learning techniques. Experiments revealed that such argumentation Boltzmann machines can outperform other classification models, especially in noisy settings.

1 Introduction

Argumentation aims at supporting rational persuasion and deliberations in domains where defeasible conclusions are raised on the basis of possibly partial and conflicting pieces of information. Studies on argumentation can be traced back to ancient times, and now comprise a focus of research in artificial intelligence (Atkinson et al. 2017) where there exists a variety of formal models to capture diverse aspects of argumentation. For instance, rule-based or logic-based argument construction models (Besnard et al. 2014) can integrate with formal approaches for the evaluation of acceptance statuses of arguments, possibly at a more abstract level (Dung 1995; Baroni, Caminada, and Giacomin 2011). Then argument acceptance can be projected to assess statement acceptance (Baroni, Governatori, and Riveret 2016). Typically, argumentation systems either identify a single skeptical outcome, or propose a set of credulous alternatives, possibly without specifying any degrees of credibility.

Probabilistic methods may be used to characterise and determine degrees of uncertainty attached to argument construction or acceptance. And indeed, the combination of formal argumentation and probability theory has received increasing attention in recent years, see e.g. (Verheij et al. 2015; Hunter and Thimm 2017; Riveret et al. 2018). Challenges in probabilistic argumentation include (i) reasoning upon the probability of argument and statement statuses (and with no particular assumptions on probabilistic dependencies), and (ii) learning the probability distribution of statuses from examples of argument or statement statuses.

To address these challenges in probabilistic argumentation, neuro-argumentative systems stand as an appealing solution. Neural networks can provide sturdy on-line learning with the possibility of massive parallel computation to learn and reason upon subtle probabilistic dependencies within a compact representation. Yet, neural networks often remain inscrutable structures of neural units unable to provide any sort of intelligible explanations to back outcomes. In that regard, arguments and their relationships could be used to bring intelligible explanatory knowledge representation and reasoning to the networks, thereby easing the communication of the learned knowledge to some agents.

Contribution. To address learning and reasoning challenges in probabilistic argumentation, as well as explainability issues in Boltzmann machines (RBMs), we propose a novel probabilistic argumentation system called neuro-symbolic argumentation machine (NSAM). We adopt probabilistic labellings in argumentation (Riveret et al. 2018), and integrate the approach with the probabilistic neural network model of RBMs to learn and reason upon distributions of argument statuses. To do so, we develop the idea of confidence rules (Tran and d'Avila Garcez 2018): prior argumentation knowledge is captured as an argumentation graph, and constraints on labellings of the graph are expressed into weighted strict disjunctive normal forms (the confidence rules) which are then incorporated into the structure of an RBM to constitute a NSAM. Given a dataset, a NSAM is trained on argument labellings of the considered argumentation graph such that any example/case of the dataset may be explained by an argument labelling.

Outline. In Sect. 2, the probabilistic argumentation framework is given. In Sect. 3, we define the general problem that we attempt to address in the paper. To address the problem, we propose NSAMs in Sect. 4 to learn distributions of labellings. The system is illustrated and evaluated in Sect. 5, and it is related to relevant works in Sect. 6,

*The original publication is available at www.kr.org.
†Joint first authors (equal contribution).

before concluding.

2 Argumentation Framework

We adopt the probabilistic semi-abstract argumentation framework taken from (Riveret et al. 2018). First, a semi-abstract argumentation framework is put forward, then the use of 'maxconsistent' labellings is proposed for our purposes, and the probabilistic development is exposed.

Semi-abstract argumentation. We assume a language, such that any statement pertaining to the language can be the conclusion of an argument; and any argument has a unique identifier to discern arguments with equal conclusions.

Definition 1. *Given a language Ψ and a set of argument identifiers $\mathcal{I}$, an **argument** is a tuple $\langle id, \phi \rangle$ where $id \in \mathcal{I}$ is the unique identifier of the argument and $\phi \in \Psi$ is the conclusion of the argument.*

Notation 1. *The conclusion of an argument $A = \langle id, \phi \rangle$ is denoted $\mathsf{con}(A)$, i.e. $\mathsf{con}(A) = \phi$.*

In the forthcoming probabilistic argumentation setting, an important postulate is that the event of an argument necessarily occurs along with the events of its subarguments. In that regard, classic abstract argumentation graphs (Dung 1995) lack a subargument relation to cater for such an assumption at an abstract level. To address this lacuna, we rely on so-called semi-abstract argumentation graphs (Riveret et al. 2018) featuring subargument and attack relations over arguments, cf. (Cayrol and Lagasquie-Schiex 2013; Dung and Thang 2014; Prakken 2014; Cohen et al. 2014) for similar settings.

Definition 2. *A **semi-abstract argumentation graph** is a tuple $\langle \mathcal{A}, \rightsquigarrow, \Rightarrow \rangle$ where $\mathcal{A}$ is a set of arguments, $\rightsquigarrow \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation of attack, and $\Rightarrow \subseteq \mathcal{A} \times \mathcal{A}$ is a binary (direct subargument) relation of support.*

Notation 2. *Given a semi-abstract argumentation graph $G = \langle \mathcal{A}, \rightsquigarrow, \Rightarrow \rangle$, we may write $\mathcal{A}$ as $\mathcal{A}_G$, $\rightsquigarrow$ as $\rightsquigarrow_G$, and $\Rightarrow$ as $\Rightarrow_G$.*

As to the terminology, a semi-abstract argumentation graph may be called a *bipolar* argumentation graph/framework, as long as the support relation is understood as a (direct) subargument relation, every argument has a conclusion which is a statement, and such bipolar graphs enjoy all upcoming (probabilistic) considerations on semi-abstract argumentation graphs, cf. (Polberg and Hunter 2018).

Some semi-abstract argumentation graphs may not appear 'well-formed' since, for instance, an argument can attack an argument without attacking its 'super-arguments'. We consider thus well-formed semi-abstract argumentation graphs.

Definition 3. *A semi-abstract argumentation graph G is a **well-formed semi-abstract argumentation graph** iff the relation $\Rightarrow_G$ is acyclic and antireflexive, and if $A \rightsquigarrow_G B$, and $B \Rightarrow_G C$, then $A \rightsquigarrow_G C$.*

In the remainder, we assume that all argumentation graphs are semi-abstract and well-formed. We will also use subgraphs to build our probabilistic setting.

Definition 4. *An argumentation graph H is a (legal) **subgraph** of an argumentation graph G induced by a set of arguments $\mathcal{A}_H \subseteq \mathcal{A}_G$ iff $H = \langle \mathcal{A}_H, \rightsquigarrow_G \cap (\mathcal{A}_H \times \mathcal{A}_H), \Rightarrow_G \cap (\mathcal{A}_H \times \mathcal{A}_H) \rangle$, and if $A \in \mathcal{A}_H$ and $(B, A) \in \Rightarrow_G$ then $B \in \mathcal{A}_H$.*

Given an argumentation graph, we can specify those arguments that are accepted or discarded. To do so, we label arguments as reviewed in (Baroni, Caminada, and Giacomin 2011), but slightly adapted to our probabilistic development. Accordingly, we distinguish $\{\mathsf{I}, \mathsf{O}, \mathsf{U}\}$-labellings and $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labellings. In a $\{\mathsf{I}, \mathsf{O}, \mathsf{U}\}$-labelling, each argument is associated with exactly one label which is either I, O, U: a label I (for 'in') means the argument is accepted while a label O ('out') indicates that it is rejected. Label U marks the argument as undecided. $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labellings integrate a label F (for 'off') to indicate that an argument is not expressed (i.e. its event does not occur).

Definition 5. *Let G be an argumentation graph. A $\{\mathsf{I}, \mathsf{O}, \mathsf{U}\}$-**labelling** ($\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labelling resp.) of G is a total function $L : \mathcal{A}_G \to \{\mathsf{I}, \mathsf{O}, \mathsf{U}\}$ ($L : \mathcal{A}_G \to \{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$ resp.).*

Notation 3. *The set of arguments labelled $l \in \{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$ in a labelling L is denoted $l(L) = \{A \mid L(A) = l\}$. Accordingly, a $\{\mathsf{I}, \mathsf{O}, \mathsf{U}\}$-labelling L is represented as a tuple $\langle \mathsf{I}(L), \mathsf{O}(L), \mathsf{U}(L) \rangle$, and a $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labelling L as a tuple $\langle \mathsf{I}(L), \mathsf{O}(L), \mathsf{U}(L), \mathsf{F}(L) \rangle$.*

We will work with complete labellings, and match any subgraph with complete $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labellings by 'switching off' arguments outside the considered subgraph.

Definition 6. *A **complete** $\{\mathsf{I}, \mathsf{O}, \mathsf{U}\}$-**labelling** of an argumentation graph G is a $\{\mathsf{I}, \mathsf{O}, \mathsf{U}\}$-labelling such that for every argument A in $\mathcal{A}_G$: A is labelled I iff all attackers of A are O, and A is labelled O iff A has an attacker labelled I.*

Definition 7. *Let H be a subgraph of an argumentation graph G. A **complete** $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-**labelling** of G wrt H is a $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labelling such that every argument in $\mathcal{A}_H$ is labelled according to a complete $\{\mathsf{I}, \mathsf{O}, \mathsf{U}\}$-labelling of H, every argument in $\mathcal{A}_G \backslash \mathcal{A}_H$ is labelled F.*

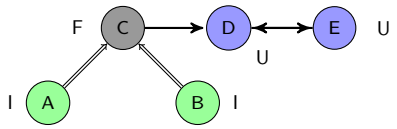


Figure 1: An argumentation graph and one of its complete $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labelling. Arguments A and B support argument C. Argument C attacks D, Arguments D and E attack each other.

A labelling of a set of statements is a function associating any statement with a label. Various specifications for statement labellings are possible, see e.g. (Baroni and Riveret 2019). For our ends, we consider the labelling which is perhaps the simplest in a meaningful way, namely a bivalent $\{\mathsf{y}, \mathsf{n}\}$-labelling, according to which a statement is either accepted (labelled y) or not (labelled n).

As to the terminology, a statement labelling may be called an example or a *case*, and will favour the latter term in the remainder.

Definition 8. *A **bivalent** $\{\mathsf{y},\mathsf{n}\}$**-labelling (or a case)** of a set of statements $\mathcal{S}$ is a total function $K : \mathcal{S} \to \{\mathsf{y},\mathsf{n}\}$.*

If statements are labelled relatively to a particular acceptance argument labellings, then we have acceptance bivalent $\{\mathsf{y},\mathsf{n}\}$-labellings, see (Baroni, Governatori, and Riveret 2016).

Definition 9. *Let $\mathcal{S}$ be a set of statements. An **acceptance bivalent** $\{\mathsf{y},\mathsf{n}\}$**-labelling** of $\mathcal{S}$ and from an argument $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labelling L is a total function $K_L : \mathcal{S} \to \{\mathsf{y},\mathsf{n}\}$ such that for any $\phi \in \mathcal{S}$: $K_L(\phi) = \mathsf{y}$ iff $\{L(A) \mid \mathsf{con}(A) = \phi\} \supseteq \{\mathsf{I}\}$, and $K_L(\phi) = \mathsf{n}$ otherwise.*

Notation 4. *A $\{\mathsf{y},\mathsf{n}\}$-labelling K may be represented as a tuple $\langle \mathsf{y}(K), \mathsf{n}(K)\rangle$ with the obvious meaning.*

Eventually, a dataset is defined as a collection of cases. Henceforth, only finite cases and finite non-empty datasets are considered, such that all the cases in a dataset are bivalent $\{\mathsf{y},\mathsf{n}\}$-labellings over the same finite non-empty set of statements.

Given a dataset and an argumentation graph, we can seek labellings of the graph which are maximally consistent with cases in the dataset, as proposed next.

Maxconsistent labellings. Given an argumentation graph and a case, none of its complete $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labellings may be consistent with the case, i.e. the case may not be an acceptance bivalent $\{\mathsf{y},\mathsf{n}\}$-labelling from any complete $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labellings. Thus we may attempt to label the graph to make it consistent with the case as much as possible. To do so, we will use so-called 'maxconsistent' $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labellings (Riveret 2020). Such labellings are obtained by labelling I every argument whose conclusion is in $\mathsf{y}(K)$, as long as all its subarguments are labelled I, as characterised by a 'maxconsistent characteristic function'.

Definition 10. *Let K be a case and G an argumentation graph. The **maxconsistent characteristic function** of G wrt K is a function $F_{G,K} : pow(\mathcal{A}_G) \to pow(\mathcal{A}_G)$ such that $F_{G,K}(\mathcal{A}) = \{A \mid \mathsf{con}(A) \in \mathsf{y}(K)$, and $\forall B \in \mathcal{A}_G$: if $B \Rightarrow_G A$ then $B \in \mathcal{A}\}$.*

Example 1. *Let G denote the argumentation graph drawn in Fig. 1, such that $\mathsf{con}(\mathsf{A}) = \mathsf{a}$, $\mathsf{con}(\mathsf{B}) = \mathsf{b}$, $\mathsf{con}(\mathsf{C}) = \mathsf{c}$, $\mathsf{con}(\mathsf{D}) = \mathsf{d}$, and $\mathsf{con}(\mathsf{E}) = \mathsf{e}$. Given the case $\mathsf{K} = \langle\{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{e}\},\{\mathsf{d}\}\rangle$, it holds that: $F_{\mathsf{G},\mathsf{K}}(\emptyset) = \{\mathsf{A},\mathsf{B},\mathsf{E}\}$ and $F^2_{\mathsf{G},\mathsf{K}}(\emptyset) = \{\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{E}\}$, and $F^3_{\mathsf{G},\mathsf{K}}(\emptyset) = F^2_{\mathsf{G},\mathsf{K}}(\emptyset)$. Thus, the function $F_{\mathsf{G},\mathsf{K}}$ has a fixed point which is the set $\{\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{E}\}$.*

A characteristic function $F_{G,K}$ is monotonic, and if an argument is included in $F^i_{G,K}(\emptyset)$, then it is also included in $F^j_{G,K}(\emptyset)$ $(i \leq j)$, thus there exists a unique fixed point $\mathcal{A}^* = F^i_{G,K}(\emptyset)$, leading us to maxconsistent $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labellings.

Definition 11. *Let K be a case, G an argumentation graph, and $\mathcal{A}^* = F^i_{G,K}(\emptyset)$ $(0 \leq i)$ the fixed point of the maxconsistent characteristic function of G wrt K. A $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labelling L of G is **maxconsistent** with K (or K-maxconsistent) iff $\mathsf{I}(L) = \mathcal{A}^*$.*

Maxconsistency of a $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labelling with a case K specifies the arguments labelled I. To address the labelling of other arguments, we may hold K-maxconsistent complete $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labellings.

Algorithm 1 Computation of a bivalent $\{\mathsf{y},\mathsf{n}\}$-labelling maxconsistent with a case.

1: **input** A case K of a set of statements $\mathcal{S}$, and an argumentation graph G.
2: $\mathsf{I}(L_0) \leftarrow \emptyset$.
3: **repeat**
4: $\quad \mathsf{I}(L_{i+1}) \leftarrow \mathsf{I}(L_i) \cup \{A \mid \mathsf{con}(A) \in \mathsf{y}(K), \text{and } \forall B \in \mathcal{A}_G : \text{if } B \Rightarrow A \text{ then } B \in \mathsf{I}(L_i)\}$
5: **until** $L_i = L_{i+1}$
6: $\mathsf{y}(K') = \{\phi \in \mathcal{S} \mid \exists A \in \mathsf{I}(L_i), \mathsf{con}(A) = \phi\}$
7: **return** $\langle \mathsf{y}(K'), \mathcal{S} \backslash \mathsf{y}(K')\rangle$.

Definition 12. *Let K be a case, and G an argumentation graph. A $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labelling L of G is a K**-maxconsistent complete** $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$**-labelling** iff L is maxconsistent with K, and L is a complete labelling of G.*

We can note that, given a case, an argumentation graph may have no complete $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labellings which are maxconsistent with the case. However, any case over a set of statements $\mathcal{S}$ has an argumentation graph for which a complete $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labelling is maxconsistent with the case, e.g. any graph $\langle \mathcal{A}, \rightsquigarrow, \emptyset\rangle$ such that every statement in $\mathcal{S}$ is the conclusion of at least one argument in $\mathcal{A}$, and such that for any argument $A, B \in \mathcal{A}$, if $\mathsf{con}(A), \mathsf{con}(B) \in \mathsf{y}(K)$ then A does not attack B (i.e. $A \not\rightsquigarrow B$). For our purposes, an explanation of a case K refers to a K-maxconsistent $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labelling of the given argumentation graph; and if there is no K-maxconsistent $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labelling which is complete then the argumentation graph may be revised, but not necessarily since the case may be simply 'corrupted'.

From any maxconsistent $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labelling, we can then define a maxconsistent bivalent $\{\mathsf{y},\mathsf{n}\}$-labelling.

Definition 13. *A bivalent $\{\mathsf{y},\mathsf{n}\}$-labelling K' of an argumentation graph G is **maxconsistent** with a case K (or K-maxconsistent) iff K' is the bivalent $\{\mathsf{y},\mathsf{n}\}$-labelling from a K-maxconsistent $\{\mathsf{I},\mathsf{O},\mathsf{U},\mathsf{F}\}$-labelling L of G.*

In practice, the bivalent $\{\mathsf{y},\mathsf{n}\}$-labelling maxconsistent with a case can be efficiently computed using Alg. 1. It begins with an empty set of arguments labelled I (line 2). Then an iteration begins. If an argument has a conclusion labelled y, and all its subarguments are labelled I, then it is labelled I (line 4). The iteration continues until no more arguments can be labelled. The algorithm terminates by labelling y any statement for which there exists an argument labelled I.

Probabilistic argumentation. The probability space for probabilistic argumentation can be defined in many ways. For our ends, we adopt the approach taken in (Riveret et al. 2018) where the sample space is a set of specific labellings of a given argumentation graph.

Definition 14. *A **probabilistic labelling frame** is a tuple $(G, (\Omega, F, P))$ such that G is an argumentation graph, and (Ω, F, P) is a probability space such that: the sample space Ω is a non-empty set of (specific) labellings of G, the σ-algebra F is the power set of Ω, i.e. $F(\Omega) = 2^\Omega$, P is the probability function from $F(\Omega)$ to $[0, 1]$.*

The proposed probabilistic setting is generic in the sense that it can accommodate various types of labellings such as complete $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labellings, preferred $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labellings (not presented here) etc. Given an argumentation graph, we will focus on a sample space as the set of $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labellings of the graph. By doing so, any case K of a dataset can be associated with a K-maxconsistent $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labelling which belongs to the sample space. Again, if there is no K-maxconsistent $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labelling of an argumentation which turns out to be complete, then the graph may not be considered as an adequate graph and may be consequently revised.

Over a probability space of probabilistic labelling frame $(G,\ (\Omega, F, P))$, we can work with random variables, i.e. functions (usually denoted by an uppercase letter as X or Y for example) from the sample space Ω into another set of elements. Accordingly, we introduce for any statement ϕ a categorical random variable K_ϕ which takes value in the set $\{\mathsf{y}, \mathsf{n}\}$. So the event $K_\phi = \mathsf{y}$ is a shorthand for the outcomes $\{L \in \Omega \mid \exists A \in \mathcal{A}_G : L(A) = \mathsf{I}, \mathsf{con}(A) = \phi\}$. These random variables are called *random labellings*.

Notation 5.

1. *We use upper boldface type to denote sets of random labellings. So* $\mathbf{K}$ *denotes a set of random labellings* $\{K_{\phi 1}, \ldots, K_{\phi n}\}$.
2. *We use lower boldface type to denote assignments to a set of random labellings, i.e. assignments of values to the variables in the set. So given a set* $\mathbf{K} = \{K_{\phi 1}, \ldots, K_{\phi n}\}$*, a possible assignment is* $\mathbf{k} = \{K_{\phi 1} = \mathsf{y}, \ldots, K_{\phi n} = \mathsf{n}\}$.
3. *An assignment* $\mathbf{k} = \{K_{\phi 1} = k_1, \ldots, K_{\phi n} = k_n\}$ *may be used to denote the assignment corresponding to a statement labelling* K *(and vice-versa), such that* $K_{\phi i} = k_i$ *iff* $K(\phi_i) = k_i$*. Eventually, we may simply say that* $\mathbf{k} = \{K_{\phi 1} = k_1, \ldots, K_{\phi n} = k_n\}$ *is an assignment of the set of statements* $\{\phi 1, \ldots, \phi n\}$.

To recap, given a dataset and an argumentation graph as prior knowledge, for every case K in a dataset, we can build a K-maxconsistent $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labelling of the graph. In this context, an explanation of a case K refers to a K-maxconsistent $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labelling of the given argumentation graph; and if there is no K-maxconsistent $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labelling which is complete then the argumentation graph may be revised. Resulting $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labellings can be attached a probability value. As we will see soon, these $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labellings can be used to train neural networks. Then bivalent $\{\mathsf{y}, \mathsf{n}\}$-labellings associated with sampled $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labellings can be employed to evaluate predictions against the input dataset.

3 Problem Definition

Let us first consider standard binary classifiers in a probabilistic setting. Given a sample space $\mathcal{X}$ and a set of labels $\{0, 1\}$, we have a distribution P of pairs (x, y) where any $x \in \mathcal{X}$ is an example and any $y \in \{0, 1\}$ is a label. The problem addressed by a probabilistic classifier is to assign, given a new example $x \in \mathcal{X}$, a probability $P(y \mid x)$ to every label $y \in \{0, 1\}$ (such that these probabilities sum to one). The problem of a 'hard' classifier is to determine, given a new example $x \in \mathcal{X}$, its most likely label $\hat{y} = \arg\max_y P(y \mid x)$.

These problems may be adapted to our probabilistic argumentation setting as follows.

Let $\mathcal{S}$ be the disjoint union of two non-empty finite sets of statements $\mathcal{S}'$ and $\mathcal{S}''$ (i.e. $\mathcal{S} = \mathcal{S}' \cup \mathcal{S}''$ and $\mathcal{S}' \cap \mathcal{S}'' = \emptyset$).

Given:

- an empirical distribution P^+ of $\{\mathsf{y}, \mathsf{n}\}$-labellings of $\mathcal{S}$,
- an assignment $\mathbf{k}'$ of $\mathcal{S}'$.

Assign\determine:

- a probability $P^-(\mathbf{k}'' \mid \mathbf{k}')$ (from a machine) to every possible assignment $\mathbf{k}''$, where $\mathbf{k}''$ is an assignment of $\mathcal{S}''$, and such that a loss function of P^+ and P^- is minimised.

Eventually, amongst all possible $\{\mathsf{y}, \mathsf{n}\}$-labellings, we may prefer to get the $\{\mathsf{y}, \mathsf{n}\}$-labelling corresponding to the most likely assignment $\mathbf{k}''$:

$$\mathbf{k}'' = \arg\max_{\mathbf{k}''} P^-(\mathbf{k}'' \mid \mathbf{k}').$$

The loss function is left unspecified in the problem definition. For our purposes, we will use a standard Kullback-Leibler divergence measuring the distance between P^- and P^+. To minimise the divergence, we will use Boltzmann machines for capturing a distribution P^- approximating P^+ within a compact graphical model maximising the entropy of P^-.

In our context and for our purposes, an argumentation graph may be also given as prior knowledge: it may be used to deal with noisy datasets, it may also be used to provide some explanations for the resulting $\{\mathsf{y}, \mathsf{n}\}$-labellings. We will further elaborate on that in the remainder of the paper.

Example 2. *Let us suppose the argumentation graph in Fig. 1 as prior knowledge and a given dataset where the conclusions of the graph are labelled* y *or* n*. Given the labelling of statements* a*,* b *and* c *such that every statement is labelled* n*, what is the probability of labellings of statements* d *and* e*? A naive approach to answer the question is to go through the given dataset and add up the frequencies of sampled statement labellings where statements* a*,* b *and* c *are labelled* n*. However, the size of the sample space of a probabilistic labelling frame* $(G, (\Omega, F, P))$ *is superior or equal to* $|2^{\mathcal{A}_G}|$*, and thus it may be computationally too costly (in space and in time) to handle all the records. For this reason, we are looking for a compact model from which we can answer queries on probabilities of labellings. Moreover, the dataset may be noisy, and thus we would like to use the argumentation graph to 'repair' aberrant cases, and also mitigate overfitting. Eventually, for every statement labelling answered by the model, we would like to have an explanation of it in terms of argument labellings.*

4 Argumentation Machines

We now introduce our neuro-symbolic argumentation model which is based on the graphical model of *restricted Boltz-*

mann machines (RBM). To encode arguments and logical constraints into an RBM, we employ the idea of confidence rules (Tran and d'Avila Garcez 2018).

Notation 6. *In this section, to mitigate discrepancy between notation of our argumentation setting and typical notation for RBMs, argument identifiers are natural numbers, and any argument $\langle n, \phi \rangle$ may be denoted by its identifier n.*

RBMs for argumentation. RBMs (Smolensky 1986) are a probabilistic model which can be well suited for representing and reasoning with arguments. An RBM for argumentation has a hidden layer of latent units h_j and a visible layer of softmax groups (a^n), each group consisting of four units corresponding to four possible labels/states $s = \{\mathsf{I}, \mathsf{O}, \mathsf{F}, \mathsf{U}\}$ of argument n. This RBM is characterised by an energy function of assignments $\mathbf{a}$ and $\mathbf{h}$ of argument states and hidden units respectively:

$$\mathrm{E}_{\mathrm{rbm}}(\mathbf{a}, \mathbf{h}) = -\sum_{njs} a_s^n w_{js}^n h_j - \sum_{ns} b_s^n a_s^n - \sum_j d_j h_j \quad (1)$$

where w_{js}^n is the connection weight between visible unit s of softmax group n to the hidden unit h_j, b_s^n is the bias for unit s of group n, and d_j is the bias for hidden unit j. Argumentation constraints are captured in such an RBM as a joint distribution $p(\mathbf{a}) = \sum_{\mathbf{h}} \exp(-\mathrm{E}_{\mathrm{rbm}}(\mathbf{a}, \mathbf{h}))/Z$, which quantify how likely is an argument labelling, and where $Z = \sum_{\mathbf{a},\mathbf{h}} \exp(-\mathrm{E}_{\mathrm{rbm}}(\mathbf{a}, \mathbf{h}))$ is the partition function. Note that computing the partition function Z is intractable.

To avoid the intractability issue, inferences in RBMs can be done using Gibbs sampling, i.e. by determining the states of units in a layer given the states of the units in the other layer. For example, let us assume that $\mathcal{A}_{\mathrm{unk}}$ is a subset of arguments whose labelling is unknown (we would like to infer this labelling), and the subset $\mathcal{A}_{\mathrm{know}} = \mathcal{A} \backslash \mathcal{A}_{\mathrm{unk}}$ consists of the rest of the arguments whose labelling is known. By performing Gibbs sampling, first we assign the states of arguments in $\mathcal{A}_{\mathrm{unk}}$ with equal probabilities, i.e. $1/4$, then we iteratively update the states of the hidden units using $p(h_j|\mathbf{a}) = \mathrm{sigmoid}(\sum_{ns} a_s^n w_{js}^n + d_j)$ and the states of unknown arguments using $p(a_s^n|\mathbf{h}) = \mathrm{sigmoid}(\sum_j w_{js}^n h_j + b_s^n)$. This process is run repeatedly as a Markov chain while clamping the states of known arguments in $\mathcal{A}_{\mathrm{know}}$ until convergence. This inference procedure is approximate.

Alternatively, we can make inferences by considering the distribution of a state assignment $\mathbf{a}^*$ of some arguments given the state assignment $\mathbf{a}^{-*}$ of other arguments.

$$p(\mathbf{a}^* \mid \mathbf{a}^{-*}) = \frac{\exp(-\mathrm{F}(\mathbf{a}^*, \mathbf{a}^{-*}))}{\sum_{\mathbf{a}^*} \exp(-\mathrm{F}(\mathbf{a}^*, \mathbf{a}^{-*}))} \quad (2)$$

where $\mathrm{F}(\mathbf{a}^*, \mathbf{a}^{-*})$ denotes the free energy for assignment $(\mathbf{a}^*, \mathbf{a}^{-*})$ such that $\mathrm{F}(\mathbf{a}^*, \mathbf{a}^{-*}) = -\sum_{ns} b_s^n a_s^n - \sum_j \log(1 + \exp(\sum_{ns} a_s^n w_j^n + d_j))$. The distribution in (2) can be computed analytically as the partition function has been canceled out. However, in general, computing the distribution is exponentially expensive over the number of unknown labelling assignments. Yet, this type of inference is useful in the case of small number of unknown arguments.

For training, the most popular method is Contrastive Divergence (Hinton 2002) which approximately maximises the log-likelihood of the joint distribution. We call it *generative* training. However, if one wants to use a deterministic training method, an option is to maximise the log-likelihood of the conditional distribution showed in (2), see (Cherla et al. 2017). This is known as *discriminative* training.

Confidence rules. To encode argumentation knowledge onto an RBM, we apply the approach of confidence rules (Tran and d'Avila Garcez 2018) according to which a set of confidence rules can be equivalently represented in an RBM such that maximising satisfiability (i.e. total confidence values of the satisfied rules) is equivalent to minimising the energy function of the RBM, i.e. to maximising the probability of an assignment.

Definition 15. *A **confidence rule** is an if-and-only-if formula of the form $c : h \leftrightarrow \bigwedge_t x_t \bigwedge_k \neg x_k$ where c is a positive real number (called confidence value), h is a positive literal (called hypothesis), and x_t and $\neg x_k$ are positive literals and negative literals respectively.*

Any propositional formula can be captured by a set of confidence rules by transforming it into a strict Disjunctive Normal Form (SDNF), i.e. a formula in disjunctive normal form which is true if, and only if, only one of its conjunctive clauses is true. Each conjunctive clause is then enclosed into a confidence rule by adding a hidden literal called hypothesis.

Confidence rules are efficiently encoded into a RBM by adding a hidden unit for each hypothesis along with 'confidence' connections to literals in the conjunction, with weights c and $-c$ for positive and negative literals in the conjunction respectively. Similarly to the connection weights between visible units and hidden units can be captured into a weight matrix (w_{js}^n), these confidence connections can be captured into a 'confidence weight matrix'. Finally, each hidden unit is assigned a bias $d_j = c \cdot (-n + \epsilon)$, where n is the number of positive literals in the conjunction and $0 < \epsilon < 1$ is a real number (see Example 3 for an illustration).

The role of the confidence value c is critical: it determines how hard or soft the constraint to be imposed. The higher the value, the higher the probability to satisfy the constraint. For example, a small value of c relaxes the logical constraint by allowing non-zero probability of assignments which do not satisfy the constraint. Eventually, optimal values can be learned from data or can be searched heuristically (as we will do in our experiments in Sect. 5) in order to minimise the loss function.

Example 3. *Suppose two distinct arguments m and n such that n attacks m. The rule 'if n is labelled I, then m is labelled either O or F' can be converted into a formula in SDNF as follows: $((a_{\mathsf{O}}^m \vee a_{\mathsf{F}}^m) \leftarrow a_{\mathsf{I}}^n) \wedge \neg(a_{\mathsf{O}}^m \wedge a_{\mathsf{F}}^m) \equiv (\neg a_{\mathsf{I}}^n \wedge \neg a_{\mathsf{O}}^m \wedge \neg a_{\mathsf{F}}^m) \vee (a_{\mathsf{O}}^m \wedge \neg a_{\mathsf{F}}^m) \vee (\neg a_{\mathsf{O}}^m \wedge a_{\mathsf{F}}^m)$, leading to three confidence rules: $c : h_1' \leftrightarrow \neg a_{\mathsf{I}}^n \wedge \neg a_{\mathsf{O}}^m \wedge \neg a_{\mathsf{F}}^m$, $c : h_2' \leftrightarrow a_{\mathsf{O}}^m \wedge \neg a_{\mathsf{F}}^m$ and $c : h_3' \leftrightarrow \neg a_{\mathsf{O}}^m \wedge a_{\mathsf{F}}^m$ where c is a confidence value.*

The rules are then encoded into an RBM with an energy $E_{\mathrm{rul}} = -h_1' \cdot c \cdot (-a_{\mathsf{I}}^n - a_{\mathsf{O}}^m - a_{\mathsf{F}}^m + 0.5) - h_2' \cdot c \cdot (a_{\mathsf{O}}^m - a_{\mathsf{F}}^m - 0.5) - h_3' \cdot c \cdot (-a_{\mathsf{O}}^m + a_{\mathsf{F}}^m - 0.5)$ so that argument labellings satisfying the logical rules have a higher probability than those violating them.

Finally, we assign each hidden unit a bias. For example, if $\epsilon = 0.5$, then the rule $c : h'_2 \leftrightarrow a^m_{\mathsf{O}} \wedge \neg a^m_{\mathsf{F}}$ is associated with a hidden unit h'_2 and connection weights for (h'_2 and a^m_{O}) and (h'_2 and a^m_{F}) equal to c and $-c$ respectively, while the bias for h'_2 is $c \cdot (-1+0.5) = -0.5 \cdot c$. A neuro-symbolic argumentation model with the constraint above is shown in Fig. 2. As a RBM, it is characterised by an energy $E_{\text{full}} = E_{\text{rbm}} + E_{\text{rul}}$.

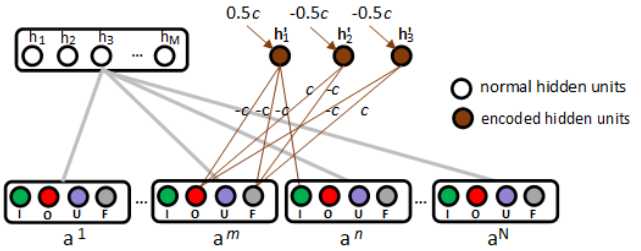


Figure 2: A neuro-symbolic argumentation RBM.

Integrating argumentation knowledge. Let us now consider argumentation constraints/rules which can be integrated in our RBM-based argumentation model.

Notation 7. *Given an argumentation graph G, and $A = \langle n, \alpha \rangle$ an argument in $\mathcal{A}_G$, the set of arguments in $\mathcal{A}_G$ attacking (supporting resp.) m is denoted $\mathcal{A}^{\rightsquigarrow m}_G$ ($\mathcal{A}^{\Rsh m}_G$ resp.).*

Propositions. Let G denote an argumentation graph, $A = \langle n, \alpha \rangle$ and $B = \langle m, \beta \rangle$ arguments in $\mathcal{A}_G$, and L a (possibly complete) $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labelling of G such that L is maxconsistent with a given case. We used the following propositions, each followed by a corresponding formula in SDNF (the right-hand side of the equivalence) where each conjunction is to be enclosed in a confidence rule.

Proposition 1 (Argumentation rule 1)**.** If *there exists an attacker A of B (i.e. $(A, B) \in \rightsquigarrow_G$) such that $A \neq B$, and A is labelled* I *(i.e. $L(A) = \mathsf{I}$)* then *B is labelled either* O *or* F *(i.e. either $L(B) = \mathsf{O}$ or $L(B) = \mathsf{F}$).*

$$((a^m_{\mathsf{O}} \vee a^m_{\mathsf{F}}) \leftarrow \bigvee_{n \in \mathcal{A}^{\rightsquigarrow m}_G} a^n_{\mathsf{I}}) \wedge \neg(a^m_{\mathsf{O}} \wedge a^m_{\mathsf{F}})$$

$$\equiv (\neg a^m_{\mathsf{O}} \wedge \neg a^m_{\mathsf{F}} \bigwedge_{n \in \mathcal{A}^{\rightsquigarrow m}_G} \neg a^n_{\mathsf{I}}) \vee (a^m_{\mathsf{O}} \wedge \neg a^m_{\mathsf{F}}) \vee (\neg a^m_{\mathsf{O}} \wedge a^m_{\mathsf{F}})$$

$m \notin \mathcal{A}^{\rightsquigarrow m}_G$.

Proposition 2 (Argumentation rule 2)**.** If *there exists an attacker A of B, and A is labelled* U*, and no attackers of B are labelled* I then *B is labelled either* U *or* F.

$$((a^m_{\mathsf{U}} \vee a^m_{\mathsf{F}}) \leftarrow \bigvee_{n \in \mathcal{A}^{\rightsquigarrow m}_G} a^n_{\mathsf{U}} \bigwedge_{l \in \mathcal{A}^{\rightsquigarrow m}_G} \neg a^l_{\mathsf{I}}) \wedge \neg(a^m_{\mathsf{U}} \wedge a^m_{\mathsf{F}})$$

$$\equiv (a^m_{\mathsf{U}} \wedge \neg a^m_{\mathsf{F}}) \vee (\neg a^m_{\mathsf{U}} \wedge a^m_{\mathsf{F}})$$

$$\vee (\neg a^m_{\mathsf{U}} \wedge \neg a^m_{\mathsf{F}} \bigwedge_{n \in \mathcal{A}^{\rightsquigarrow m}_G} \neg a^n_{\mathsf{U}} \bigwedge_{l \in \mathcal{A}^{\rightsquigarrow m}_G} \neg a^l_{\mathsf{I}})$$

$$\bigvee_{l_i \in \mathcal{A}^{\rightsquigarrow m}_G} (\neg a^m_{\mathsf{U}} \wedge \neg a^m_{\mathsf{F}} \wedge a^{l_i}_{\mathsf{I}} \bigwedge_{l_j \in \mathcal{A}^{\rightsquigarrow m}_G (j>i)} \neg a^{l_j}_{\mathsf{I}})$$

with $\mathcal{A}^{\rightsquigarrow m}_G = \{l_1, \ldots, l_n\}$.

Proposition 3 (Argumentation rule 3)**.** If *there exists a supporter A of B, and A is labelled* F then *B is labelled* F.

$$a^m_{\mathsf{F}} \leftarrow \bigvee_{n \in \mathcal{A}^{\Rsh m}_G} a^n_{\mathsf{F}} \equiv (\neg a^m_{\mathsf{F}} \bigwedge_{n \in \mathcal{A}^{\Rsh m}_G} \neg a^n_{\mathsf{F}}) \vee a^m_{\mathsf{F}}$$

Given an argumentation graph G, the encoded argumentation rules hold for any argument in the set of arguments $\mathcal{A}_G$. Consequently, they have to be instantiated wrt the given argumentation graph.

These argumentation rules constrain the labelling of any argument on the basis of its attackers and supporters. The first two rules specify the labelling I or F, and U or F for an argument given the labelling of its attackers. These rules are not deterministic, because the labelling of an argument is not fully determined by the labelling of its attackers and supporters. For example, an argument with no attackers nor supporters can be labelled labelled I or F. The last rule further constrains an argument to be labelled F if a supporter is labelled F, and thus it is a deterministic rule.

We have to remark that, towards completeness, other rules may/should be considered. For example, one may consider the following rule: *if* every attacker of a (non self-attacking) argument B is labelled O or F *then* B is labelled either I or F. Encoding other argumentation rules into SDNFs is left to future investigations.

In general, encoding rules is NP-complete in the worst case due to the complexity of converting logical constraints into SDNFs. Fortunately, in practice the proposed argumentation constraints are in the form of logical implications whose conversion into SDNFs has linear time complexity over the total number of literals. Argumentation rules 1, 2, 3 can be then encoded for all arguments into a confidence weight matrix, that we may call an 'argumentation weight matrix'.

Proposition 4. *Given an argumentation graph G, the space complexity of the argumentation weight matrix (encoding the argumentation rules 1, 2, 3) is $\mathcal{O}(|\mathcal{A}_G|^2)$. The time complexity of computing/generating the argumentation weight matrix is $\mathcal{O}(|\mathcal{A}_G|^2)$.*

Proof. There are three conjunctions in the formula in DNF corresponding to Argumentation rule 1. Thus three hidden units are needed to encode Argumentation rule 1 into an RBM. Similarly, the numbers of hidden units to encode argumentation rules 2 and 3 for an argument m are $3+|\mathcal{A}^{\rightsquigarrow m}_G|$ and 2, respectively. Therefore, the overall encoding results in an argumentation weight matrix of $\mathcal{O}(|\mathcal{A}_G|^2)$ elements (in the worst case where any argument has all arguments as attackers). Concerning time complexity, argumentation rules 1 and 2 for an argument m need $\propto |\mathcal{A}^{\rightsquigarrow m}_G|$ and $\propto |\mathcal{A}^{\rightsquigarrow m}_G|^2$ steps respectively to encode all literals into the RBM, while Argumentation rule 3 needs $\propto |\mathcal{A}^{\Rsh m}_G|$ steps. Therefore, the time complexity of the encoding is $\mathcal{O}(|\mathcal{A}_G|^2)$. □

Hence, for our purposes, the step of encoding argumentation rules in an RBM can be performed efficiently.

5 Experimental Evaluation

Like RBMs, NSAMs can be used for various tasks. In this paper, NSAMs are evaluated by comparing it with other common machine learning techniques at a probabilistic classification task.

5.1 General Setting

The baseline dataset stems from 2,400 routine cases as proposed in (Bench-Capon 1993) about, without quotes, a fictional welfare benefit paid to pensioners to defray expenses for visiting a spouse in hospital. The conditions to obtain a benefit (represented by statement grant) are as follows.

1. The person should be of pensionable age (60 for a woman, 65 for a man) (age), and
2. the person should have four out of the last five paid contributions in relevant contribution years (contrib), and
3. the person should be a spouse of the patient (spouse), and
4. the person should not be absent from the UK (uk), and
5. the person should have capital resources not amounting to more than $3,000$ (capital), and
6. if the relative is an in-patient (inPat) the hospital should be within a certain distance (inDist); if an out-patient (outPat), beyond that distance (outDist).

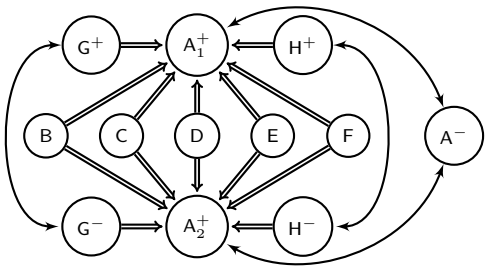
The original data (Bench-Capon 1993) is real or Boolean, whereas our setting deals with bivalent features only. For this reason, attribute-value pairs were mapped onto statements as given in the above items (e.g. age stands for a person of pensionable age). All the statements were considered to be premises, except grant and ¬grant; and grant was the target. Premise statements also included 52 'noise' statements, i.e. statements playing no role in the labelling of the target.

As we are interested in predicting non-deterministic outcomes, the baseline data set was further tuned into noisy data sets: for each noisy data set and every case of the data set, a *noise statement* was labelled n with a probability $p_{\mathbf{noise}}$, and the welfare (target variable) was set to *not granted*, even though it was *granted* in the baseline data, with a probability $p_{\mathbf{error}}$. In more detail, for every case in the baseline dataset:

- for statement grant of the case, we drew a number n between 0 and 1 from a uniform distribution: if $n < p_{\mathbf{error}}$ and the statement was labelled y then the label of the statement was changed to n; otherwise, the label of the statement was left unchanged. If statement grant was labelled y (n resp.) then statement ¬grant was labelled n (y resp.);
- for any noise statement of the case, we drew a number n between 0 and 1 from a uniform distribution: if $n < p_{\mathbf{noise}}$ then the noise statement was labelled y; else the statement was labelled n.

For any noisy data set, any evaluation was carried out by using 5-fold cross-validation, with each fold consisting of a training set and a test set. We used 60% of the data samples for training, and 40% for validation.

The prior knowledge, i.e. the conditions for a welfare benefit, was captured in an argumentation graph induced from the dataset (Riveret 2020). It is shown in Fig. 3. Each statement was the conclusion of an argument, and each noise statement was the conclusion of a 'noise argument' (not shown in Fig. 3) such that no noise argument was involved in any attack or support relationship.



$\mathsf{con}(\mathsf{A}_1^+) = \mathsf{grant}$; $\mathsf{con}(\mathsf{A}_2^+) = \mathsf{grant}$;
$\mathsf{con}(\mathsf{A}^-) = \neg\mathsf{grant}$; $\mathsf{con}(\mathsf{B}) = \mathsf{age}$;
$\mathsf{con}(\mathsf{C}) = \mathsf{contrib}$; $\mathsf{con}(\mathsf{D}) = \mathsf{spouse}$;
$\mathsf{con}(\mathsf{E}) = \mathsf{uk}$; $\mathsf{con}(\mathsf{F}) = \mathsf{capital}$;
$\mathsf{con}(\mathsf{G}^+) = \mathsf{inPat}$; $\mathsf{con}(\mathsf{G}^-) = \mathsf{outPat}$;
$\mathsf{con}(\mathsf{H}^+) = \mathsf{inDist}$; $\mathsf{con}(\mathsf{H}^-) = \mathsf{outDist}$.

Figure 3: Argumentation graph. Attacks $\mathsf{G}^+ \rightsquigarrow \mathsf{A}_2^+$, $\mathsf{H}^+ \rightsquigarrow \mathsf{A}_2^+$, $\mathsf{G}^- \rightsquigarrow \mathsf{A}_1^+$ and $\mathsf{H}^- \rightsquigarrow \mathsf{A}_1^+$ are not drawn for the sake of clarity, and noise arguments are not shown due to the lack of space.

For every case K in the dataset, we built a K-maxconsistent $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labelling of the graph. The resulting collection of $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labellings was used to train the NSAM RBM. Then bivalent $\{\mathsf{y}, \mathsf{n}\}$-labellings associated with sampled $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labellings were used to evaluate predictions against the given test data.

We compared our argumentation model with other models trained on the cases, including standard neural networks (NN), restricted Boltzmann machines (RBM), logistic regression (LR), linear discriminant (LD), quadratic discriminant (QD), decision trees (DT) and random forests (RF). We tested three different NSAM networks: $\mathrm{NSAM}_{\mathrm{rul}}$, $\mathrm{NSAM}_{\mathrm{rbm}}$ and $\mathrm{NSAM}_{\mathrm{full}}$. $\mathrm{NSAM}_{\mathrm{rbm}}$ is a standard RBM, i.e. an RBM with hidden units and softmax groups trained on the available data. $\mathrm{NSAM}_{\mathrm{rul}}$ is built from the encoded argumentation rules only. Hence, compared to $\mathrm{NSAM}_{\mathrm{rbm}}$, $\mathrm{NSAM}_{\mathrm{rul}}$ relies exclusively on the argumentation rules and the quality of such prior knowledge. $\mathrm{NSAM}_{\mathrm{full}}$ is the complete system integrating $\mathrm{NSAM}_{\mathrm{rul}}$ and $\mathrm{NSAM}_{\mathrm{rbm}}$ by training $\mathrm{NSAM}_{\mathrm{rul}}$ on the available data (similarly to Example 3).

For $\mathrm{NSAM}_{\mathrm{rbm}}$, the training set was used for model selection, i.e. for the selection of the best learning rate and the number of hidden units. In $\mathrm{NSAM}_{\mathrm{rul}}$, we attached a confidence value to each argumentation rule, and all the confidence rules induced from an argumentation rule were associated with a shared confidence value. We then performed a brute-force search for the confidence value using the validation set (the training set was not used). For training $\mathrm{NSAM}_{\mathrm{full}}$ we first found the best confidence values for the encoded part by performing a brute-force search similar as for $\mathrm{NSAM}_{\mathrm{rul}}$. Then we trained the other part (not encoded by rules) while keeping the encoded part fixed. $\mathrm{NSAM}_{\mathrm{rbm}}$ and $\mathrm{NSAM}_{\mathrm{rul}}$ systems (and thus their integration into $\mathrm{NSAM}_{\mathrm{full}}$) were trained discriminatively by maximising the log-likelihood of conditional distribution $p(\mathbf{a}^* \mid$

$\mathbf{a}^{-*}$) from Eq. (2) where $\mathbf{a}^*$ is any state assignment over arguments A_1^+, A_2^+ and A^-.

5.2 Experiments

In a first stage, we made an evaluation of NSAMs when the argumentation graph fits well the dataset, i.e. when, for or every case K in the considered dataset, the graph has a K-maxconsistent $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labelling which is complete.

The evaluation at this stage was carried out with a noisy dataset where $p_{\mathbf{noise}} = 0.6$ and $p_{\mathbf{error}} = 0.6$. We evaluated the models with prediction accuracy, F1 score (because the data was imbalanced), and log loss (since our NSAM model is a probabilistic model). The results are in Table 1.

As evidenced in Table 1, NSAMs performed better than the baselines. Among the baselines, we found that NN with two hidden layer was the best, however, adding more layers did not improve the results. In terms of log loss, implemented QD and DT returned hard predictions of the outputs instead of probabilities, therefore the log losses in these cases were very high. Overall, we found that $\text{NSAM}_{\text{full}}$ performed better than NSAM_{rul} which did better than NSAM_{rbm}. This result shows the goodness of argumentation confidence rules while keeping free weights to refine predictions.

A reason why $\text{NSAM}_{\text{full}}$ performed slightly better than NSAM_{rul} holds in that confidence rules do not cover all possible labelling features of the probabilistic framework. For example, the argumentation rule 3 can enforce that an argument is labelled F when one of its subargument is labelled F, however, the system has no confidence rules to constrain the labelling of an argument as F when all its subarguments are labelled I for example. In such cases, free weights have a role to play to refine predictions.

In a second stage of the evaluation, we evaluated $\text{NSAM}_{\text{full}}$ when the argumentation graph does not fit well the dataset, i.e. when it may have no complete $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labellings for some cases in the dataset.

To do so, we injected a 'swap noise' into the training sets of a control noisy dataset where $p_{\mathbf{noise}} = 0.5$ and $p_{\mathbf{error}} = 0.3$. The swap noise is measured through a probability value $p_{\mathbf{swap}}$. For any case of the training set of the control dataset and for any statement of the case (except statement $\neg$grant), we drew a number n between 0 and 1 from a uniform distribution: if $n < p_{\mathbf{swap}}$ then the label of the statement was left untouched; otherwise the label of the statement was swapped. Hence, when the swap noise was set at 0, all statements had their label untouched; and when the swap noise was set at 1, all statements had their label swapped. If statement grant was labelled y (n resp.) then the statement $\neg$grant was labelled n (y resp.).

Increasing levels of noise were injected into the training sets, resulting in a decrease in performance of machine learning models, see Fig. 4. In the case of $\text{NSAM}_{\text{full}}$, thanks to the encoded rules, the negative effect of the swap noise could be mitigated. For example, when the noise level was set at 1, the accuracy of all models dropped dramatically below 70% while any $\text{NSAM}_{\text{full}}$ maintained its accuracy above 80%. When the noise level reached 0.7, the accuracy of $\text{NSAM}_{\text{full}}$ was at least 25% higher than the other models.

An explanation of NSAMs' outperformance can be sketched. A NSAM uses both rules and free weights: the free weights are updated during the training, and at the beginning of the training on noisy data, such noise can start to affect the performance; however, right at the beginning, with the rules as constraints, the effect is less severe because the rules can counterbalance aberrant updates of free weights. Furthermore, in the training phase, if there is nothing useful to learn (the performance on a validation set does not increase or starts to decrease), then the training can be stopped (this is called early stopping). Accordingly, with higher noise, the training was stopped sooner to prevent a decrease of performances due to aberrant updates of the free weights. The combination of the use of confidence rules and early stopping resulted in NSAMs' outperformance for prediction purposes.

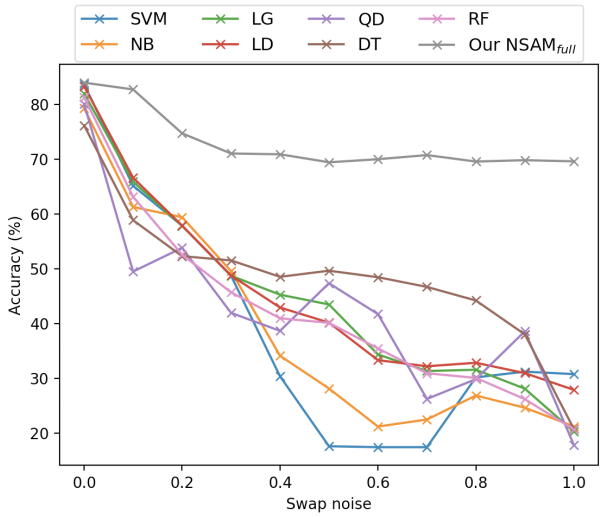


Figure 4: Effect of swap noise.

Finally, as NSAMs are trained on argument labellings instead of statement labellings, predictions can be explained in NSAMs by looking at argument labellings. For example, let us suppose a NSAM which determines that the grant is not accorded in a case where the amount of capital resources is more than 3,000. Why? Referring to Fig. 3, in that case, we can understand that argument D is labelled F in any complete $\{\mathsf{I}, \mathsf{O}, \mathsf{U}, \mathsf{F}\}$-labelling which is maxconsistent with the case (and this argument label is clamped to the machine). Consequently, we have that arguments A_1^+ and A_2^+ are labelled F, and thus statement grant is labelled n. This graph labelling explains why the grant has not been accorded. Such an operation can be performed for any (predicted) statement labellings, and thus argument labellings can be put forward to explain cases and related predictions.

6 Related Work

The work relates to research regarding neuro-symbolic systems (d'Avila Garcez, Lamb, and Gabbay 2009), and neuro-

	accuracy	F1	log loss
$\mathrm{NN}_{\mathrm{1layer}}$	80.07 ± 0.17	57.44 ± 0.70	0.36172 ± 0.00135
$\mathrm{NN}_{\mathrm{2layers}}$	80.16 ± 0.24	57.3 ± 0.45	0.36001 ± 0.00089
RBM	79.20 ± 0.48	53.77 ± 1.01	0.36525 ± 0.00442
LG	78.54 ± 0.00	58.77 ± 0.00	0.36273 ± 0.00000
LD	78.96 ± 0.00	53.94 ± 0.00	0.38460 ± 0.00000
QD	70.83 ± 0.00	64.52 ± 0.00	9.34930 ± 0.00000
DT	75.84 ± 0.21	61.73 ± 0.37	8.41461 ± 0.09637
RF	76.97 ± 0.31	58.28 ± 0.42	0.47724 ± 0.01842
$\mathrm{NSAM}_{\mathrm{rul}}$	80.79 ± 0.00	$\mathbf{65.72 \pm 0.00}$	0.33924 ± 0.00000
$\mathrm{NSAM}_{\mathrm{rbm}}$	80.45 ± 0.23	59.52 ± 0.25	0.35394 ± 0.00115
$\mathrm{NSAM}_{\mathrm{full}}$	$\mathbf{81.15 \pm 0.11}$	$\mathbf{65.86 \pm 0.21}$	$\mathbf{0.33142 \pm 0.00388}$

Table 1: Results significant with 95% confidence interval.

symbolic argumentation systems have been investigated previously in (d'Avila Garcez, Gabbay, and Lamb 2014; Riveret et al. 2015a; Riveret et al. 2015b). Different from other neuro-symbolic approaches (d'Avila Garcez, Gabbay, and Lamb 2014; Evans and Grefenstette 2018; Cohen, Yang, and Mazaitis 2017) which employ variants of feed-forward neural networks for Horn clauses, we use RBMs with confidence rules to support probabilistic reasoning on more complex logical formulas. Also, with confidence rules, we can encode our proposed argumentation rules onto an RBM, instead of learning from positive and negative assignments as in Logic Tensor Nets (Serafini and d'Avila Garcez 2016; Donadello, Serafini, and d'Avila Garcez 2017). Compared to neuro-symbolic undertakings in argumentation which have used RBMs in (Riveret et al. 2015a; Riveret et al. 2015b), our work is essentially different in that argumentation knowledge and argument labelling constraints are here incorporated within the network.

Besides neuro-symbolic argumentation systems, various systems have been investigated to determine the probability of some argument statuses given the statuses of some other arguments or premises, with respect to sundry probabilistic settings (often with no learning abilities), see e.g. (Riveret et al. 2007; Fazzinga, Flesca, and Parisi 2016; Potyka 2019; Mantadelis and Bistarelli 2020). Usually such works have strong assumptions on probabilistic dependencies (typically arguments are assumed to be probabilistically independent) while we do not rely on such assumptions thanks to the probabilistic graphical model of RBMs.

Beyond argumentation systems, logics and principled probabilistic approaches along with machine learning have been largely studied to learn from uncertain knowledge and to perform inferences with this knowledge (Getoor and Taskar 2007). Probabilistic dependencies amongst logical statements are typically correlated by design to logical dependencies, see e.g. (Richardson and Domingos 2006), whereas, in our approach, such an assumption is relaxed. In addition, by using an argumentation framework, we aim at paving the way to the use of fined-grained acceptance labellings (Baroni and Riveret 2019) for systems combining logics and principled probabilistic approaches.

The work reported here can be naturally located in the field of explainable artificial intelligence where the results of the solutions are meant to be intelligible by the concerned human agents. Arguably, the question 'What is a good explanation?' remains quite elusive, see diverse conceptions in philosophy e.g. (Hempel and Oppenheim 1948; Lipton 2016; Bechtel and Abrahamsen 2005), psychology (Keil 2005; Lombrozol 2006) or (explainable) artificial intelligence (Lipton 2016; Freitas 2014; Doshi-Velez and Kim 2017; Doran, Schulz, and Besold 2017). Our assumption is that argumentation graphs and associated labellings can be used to bring intelligible explanatory knowledge representation and reasoning to neural networks.

7 Conclusion

We have proposed neural networks which are trained on data explanations understood as argumentation graph labellings, so that any outcome is associated with an explanation.

The approach has been applied to a novel neuro-symbolic model where neural networks are RBMs and the symbolic formalism relies on probabilistic semi-abstract argumentation. Any dataset is conceived as a collection of statement labellings of an argumentation graph (the prior knowledge). Then, in place of training on statement labellings, the network is trained on corresponding argument labellings of the graph. Hence, instead of training the network on given data, we have proposed to train the network on explanations seen as argument labellings of a graph. Then we have proposed to incorporate labelling constraints into the machines, so that the sampling space of argument labellings is constrained.

Eventually, experiments have revealed that such argumentation Boltzmann machines can outperform other standard classification models, especially in noisy settings.

The proposed system is not without defaults. In particular, as mentioned earlier, the proposed argument labelling semantics clearly explain when an argument is labelled F when one of its subargument is labelled F, however, the system provides no logical explanations for a F-labelled argument for which all its subarguments are labelled I. Furthermore, due to the limitations of the framework of confidence rules in terms of strict DNF, we did not encode some possible constraints from labelling semantics. Encoding such other constraints is left to future research, and it would be interesting to investigate whether corresponding confidence rules could further improve prediction results.

References

Atkinson, K.; Baroni, P.; Giacomin, M.; Hunter, A.; Prakken, H.; Reed, C.; Simari, G. R.; Thimm, M.; and Villata, S. 2017. Towards artificial argumentation. *AI Magazine* 38(3):25–36.

Baroni, P., and Riveret, R. 2019. Enhancing statement evaluation in argumentation via multi-labelling systems. *J. Artificial Intelligence Research* 66:793–860.

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

Baroni, P.; Governatori, G.; and Riveret, R. 2016. On labelling statements in multi-labelling argumentation. In *Proc. of the 22nd Eur. Conf. on Artificial Intelligence*, 489–497. IOS Press.

Bechtel, W., and Abrahamsen, A. 2005. Explanation: A mechanist alternative. *Studies in History and Philosophy of Biol and Biomed Sci* 36(2):421–441.

Bench-Capon, T. 1993. Neural networks and open texture. In *Proc. of the 4th Int. Conf. on Artificial Intelligence and Law*, 292–297. ACM.

Besnard, P.; García, A. J.; Hunter, A.; Modgil, S.; Prakken, H.; Simari, G. R.; and Toni, F. 2014. Introduction to structured argumentation. *Argument & Computation* 5(1):1–4.

Cayrol, C., and Lagasquie-Schiex, M. 2013. Bipolarity in argumentation graphs: Towards a better understanding. *Int. J. Approximate Reasoning* 54(7):876–899.

Cherla, S.; Tran, S. N.; d'Avila Garcez, A. S.; and Weyde, T. 2017. Generalising the discriminative restricted boltzmann machines. In *Proc. of the 26th Int. Conf. on Artificial Neural Networks*, 111–119. Springer.

Cohen, A.; Gottifredi, S.; Garcia, A. J.; and Simari, G. R. 2014. A survey of different approaches to support in argumentation systems. *Knowledge Engineering Review* 29(5):513–550.

Cohen, W. W.; Yang, F.; and Mazaitis, K. 2017. Tensorlog: Deep learning meets probabilistic dbs. *CoRR* abs/1707.05390.

d'Avila Garcez, A.; Gabbay, D. M.; and Lamb, L. C. 2014. A neural cognitive model of argumentation with application to legal inference and decision making. *J. Applied Logic* 12(2):109 – 127.

d'Avila Garcez, A.; Lamb, L. C.; and Gabbay, D. M. 2009. *Neural-Symbolic Cognitive Reasoning*. Springer.

Donadello, I.; Serafini, L.; and d'Avila Garcez, A. 2017. Logic tensor networks for semantic image interpretation. In *Proc. of the 26th Int. Joint Conf. on Artifical Intelligence*, 1596–1602.

Doran, D.; Schulz, S.; and Besold, T. R. 2017. What does explainable AI really mean? A new conceptualization of perspectives. *CoRR* abs/1710.00794.

Doshi-Velez, F., and Kim, B. 2017. Towards a rigorous science of interpretable machine learning. *CoRR* abs/1702.08608.

Dung, P. M., and Thang, P. M. 2014. Closure and consistency in logic-associated argumentation. *J. Artificial Intelligence Research* 49(1):79–109.

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

Evans, R., and Grefenstette, E. 2018. Learning explanatory rules from noisy data. *J. Artificial Intelligence Research* 61:1–64.

Fazzinga, B.; Flesca, S.; and Parisi, F. 2016. On efficiently estimating the probability of extensions in abstract argumentation frameworks. *Int. J. Approximate Reasoning* 69:106 – 132.

Freitas, A. A. 2014. Comprehensible classification models: A position paper. *SIGKDD Explorations Newsletter* 15(1):1–10.

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

Hempel, C. G., and Oppenheim, P. 1948. Studies in the logic of explanation. *Philosophy of Science* 15(2):135–175.

Hinton, G. E. 2002. Training products of experts by minimizing contrastive divergence. *Neural Comput.* 14(8):1771–1800.

Hunter, A., and Thimm, M. 2017. Probabilistic reasoning with abstract argumentation frameworks. *J. Artificial Intelligence Research* 59:565–611.

Keil, F. 2005. Explanation and understanding. *Annual review of psychology* 57:227–254.

Lipton, Z. C. 2016. The mythos of model interpretability. *CoRR* abs/1606.03490.

Lombrozol, T. 2006. The structure and function of explanations. *Trends in Cognitive Sciences* 10(10):464–470.

Mantadelis, T., and Bistarelli, S. 2020. Probabilistic abstract argumentation frameworks, a possible world view. *Int. J. Approximate Reasoning* 119:204–219.

Polberg, S., and Hunter, A. 2018. Empirical evaluation of abstract argumentation: Supporting the need for bipolar and probabilistic approaches. *Int. J. Approximate Reasoning* 93:487–543.

Potyka, N. 2019. A polynomial-time fragment of epistemic probabilistic argumentation. In *Proc. of the 18th Int. Conf. on Autonomous Agents and MultiAgent Systems*, 2165–2167. IFAAMAS.

Prakken, H. 2014. On support relations in abstract argumentation as abstractions of inferential relations. In *Proc, of the 21st Eur. Conf. on Artificial Intelligence*, 735–740. IOS Press.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine Learning* 62(1-2):107–136.

Riveret, R.; Rotolo, A.; Sartor, G.; Prakken, H.; and Roth, B. 2007. Success chances in argument games: a probabilistic approach to legal disputes. In *Proc. of the 20th Annual*

Conf. on Legal Knowledge and Information Systems Legal Knowledge and Information Systems, 99–108. IOS Press.

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

Riveret, R.; Pitt, J. V.; Korkinof, D.; and Draief, M. 2015b. Neuro-symbolic agents: Boltzmann machines and probabilistic abstract argumentation with sub-arguments. In *Proc. of the 14th Int. Conf. on Autonomous Agents and Multiagent Systems*, 1481–1489. ACM.

Riveret, R.; Baroni, P.; Gao, Y.; Governatori, G.; Rotolo, A.; and Sartor, G. 2018. A labelling framework for probabilistic argumentation. *Ann. Math. Artificial Intelligence* 83(1):21–71.

Riveret, R. 2020. On searching explanatory argumentation graphs. *J. of Applied Non-Classical Logics* 30:1–70.

Serafini, L., and d'Avila Garcez, A. 2016. Learning and reasoning with logic tensor networks. In *Proc. of the 15th Int. Conf. of the Italian Association for Artificial Intelligence*, 334–348.

Smolensky, P. 1986. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1. MIT Press. chapter Information Processing in Dynamical Systems: Foundations of Harmony Theory, 194–281.

Tran, S., and d'Avila Garcez, A. 2018. Deep logic networks: Inserting and extracting knowledge from deep belief networks. *IEEE T. Neur. Net. Learning Syst.* 29(29):246–258.

Verheij, B.; Bex, F.; Timmer, S. T.; Vlek, C. S.; Meyer, J.-J. C.; Renooij, S.; and Prakken, H. 2015. Arguments, scenarios and probabilities: connections between three normative frameworks for evidential reasoning. *Law, Probability and Risk* 15(1):35–70.