Neuro-Symbolic Agents: Boltzmann Machines and Probabilistic Abstract Argumentation with Sub-Arguments

Régis Riveret
Imperial College London
London, United Kingdom
r.riveret@imperial.ac.uk

Jeremy Pitt
Imperial College London
London, United Kingdom
j.pitt@imperial.ac.uk

Dimitrios Korkinof
Cortexica Vision Systems
London, United Kingdom
dimitrios.korkinof@cortexica.com

Moez Draief
Math. and Alg. Sciences Lab
France Research Center
Huawei Technologies Co. Ltd.
moez.draief@huawei.com

ABSTRACT

Towards neuro-argumentative agents based on the seamless integration of neural networks and defeasible formalisms, with principled probabilistic settings and along efficient algorithms, we investigate argumentative Boltzmann machines where the possible states of a Boltzmann machine are constrained by a prior argumentative knowledge. To make our ideas as widely applicable as possible, and acknowledging the role of sub-arguments in probabilistic argumentation, we consider an abstract argumentation framework accounting for sub-arguments, but where the content of (sub-)arguments is left unspecified. We validate our proposal with artificial datasets and suggest its advantages.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Learning, Connectionism and neural nets, Knowledge Acquisition

General Terms

Algorithms, Theory

Keywords

Machine Learning, Neuro-Symbolic Systems, Probabilistic Argumentation, Graphical Models, Boltzmann Machines.

1. INTRODUCTION

Neuro-symbolic agents join the strength of neural network models and logics [8]: neural networks offer sturdy on-line learning with the possibility of massive parallel computation, while logic brings intelligible knowledge representation and reasoning into the networks with explanatory ability, thereby facilitating the transmission of learned knowledge to other agents. Considering the importance of defeasible reasoning and the intuitive account of this type of reasoning by formal argumentation for qualitative matters, neuro-argumentative agents shall provide an ergonomic representation and argumentative reasoning aptitude to the underlying neural networks and ease its possible combination with some argumentative agent communication languages. However, such neuro-argumentative agents have been hardly investigated so far, c.f. [3].

A parallel line of research regards the seamless integration of defeasible reasoning and probabilistic settings to capture qualitative as well as quantitative uncertainty. Several frameworks for probabilistic argumentation have been proposed in the last few years, see e.g. [13, 15, 24, 4, 7] with applications suggested in the legal domain [20, 6]. The underlying argumentation frameworks and attached probability spaces of these approaches vary in form or content, but common issues regard the assumption of independent arguments or rules, the large sample spaces and the complexity of computing probabilistic status of arguments. Moreover, they typically rely on prior probability values given by some operators, that may not be convenient for objective applications. So, efficient approaches for reasoning and learning along a probabilistic setting are solicited.

Contribution. Towards the construction of neuro-argumentative agents based on the seamless combination of neural networks and formal argumentation with principled probabilistic settings along efficient algorithms for on-line learning and reasoning, we investigate a class of Boltzmann machines called *argumentative Boltzmann machines*. To do so, an energy-based probabilistic framework for abstract argumentation with supports, akin to the standard exponential family is proposed so that the assumption of independent argument is relaxed. Then we investigate an efficient algorithm integrated to Boltzmann machines to learn a probability distribution of labellings from a set of examples. Our proposal is limited to the epistemic apparatus of agents, but it shall also inform practical mechanisms towards action. We focus on grounded semantics [5], and work on the probabilistic distribution of argument labellings [1]: given an argumentation framework, probability measures the likeliness that the event of a labelling occurs. Though such a probabilistic setting involves a priori an explosion of the

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

computational complexity, we show that it can be dramatically contained with Boltzmann machines. As proposed by Mozina et al. [18] and D'Avila Garcez et al. [3], argumentation appears as a guide constraining the space of hypotheses. Learning the relation of attacks or support amongst arguments is not addressed in this paper.

Outline. Our setting for probabilistic argumentation is introduced in Section 2. We overview Boltzmann machines in Section 3 and we propose "argumentative Boltzmann machines" in Section 4, with experimental insights in Section 5 before concluding.

2. PROBABILISTIC ARGUMENTATION

When sub-arguments appear in an argumentation framework, a sound probabilistic setting requires to cater for these relations. For this reason, we extend the common definition of an argumentation graph of Dung [5] with the relation of support. Here support is a relation of sub-argument given that arguments "do not simply stand for statements but encode the way in which they are derived from previous statements" as stated in [19].

Definition 1. [Argumentation graph] An *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 relation of support such that, if an argument A supports B and C attacks A, then C attacks B.

As for notation, given a graph $G = \langle \mathcal{A}, \rightsquigarrow, \Rightarrow \rangle$, we write $\mathcal{A}_G = \mathcal{A}$. This setting has the advantage of its simplicity and its straightforward instantiation into common rule-based argumentation frameworks (e.g. ASPIC+ [17, 19] or DL [9]). However, notice that this notion of support is thus different from possible interpretations found in other frameworks such as in Bipolar Argumentation [2, 19].

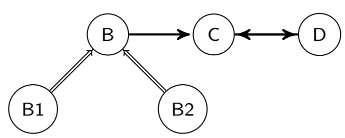


Figure 1: An argumentation graph. The argument B attacks C, the arguments C and D attack each others. The arguments B1 and B2 are sub-arguments of the argument B.

Given an argumentation graph, we can compute sets of justified or discarded arguments, i.e. arguments that shall survive or not to attacks. To do that, we will label arguments as in [1], but slightly adapted to our probabilistic setting. Accordingly, we will distinguish three labellings: $\{\mathsf{on}, \mathsf{off}\}$-labelling, $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling and $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling. In a $\{\mathsf{on}, \mathsf{off}\}$-labelling, each argument is associated with one label which is either on or off to indicate whether an argument is expressed or not (i.e. the event of an argument occur or not). In a $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling, each argument is associated with one label which is either in, out, un: a label "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. The $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling extends a $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling with the off label to indicate that an argument is not expressed (i.e. its event does not occur).

Definition 2. Let G be an argumentation graph.

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

As for notation, the sets of arguments labelled by on, in, out, un or off are respectively denoted as $\mathsf{on}(L) = \{A | L(A) = \mathsf{on}\}$, $\mathsf{in}(L) = \{A | L(A) = \mathsf{in}\}$, $\mathsf{out}(L) = \{A | L(A) = \mathsf{out}\}$, $\mathsf{un}(L) = \{A | L(A) = \mathsf{un}\}$, and $\mathsf{off}(L) = \{A | L(A) = \mathsf{off}\}$. A $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling L will be represented as a tuple $(\mathsf{in}(L), \mathsf{out}(L), \mathsf{un}(L))$, and a $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling L as a tuple $(\mathsf{in}(L), \mathsf{out}(L), \mathsf{un}(L), \mathsf{off}(L))$. Next we define the set of *complete labellings* of an argumentation graph to account for some constraints amongst the labels in and out.

Definition 3. [Complete $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling] A *complete* $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-*labelling* of an argumentation graph G is a $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling such that for every argument A in $\mathcal{A}_G$ it holds that:

- A is labelled in if and only if all attackers of A are out,
- A is labelled out if and only if A has an attacker in.

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 (or equivalently with the largest set of labels un) [5, 16] called the grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling.

Definition 4. [Grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling] A *grounded* $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-*labelling* L of an argumentation graph G is a complete $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling of G such that $\mathsf{in}(L)$ is minimal (w.r.t. set inclusion) among all complete $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labellings of G.

An algorithm for generating the grounded labelling of an argumentation graph is given in Algo. 1 (see [16]). It begins by labelling in all arguments not being attacked or whose attackers are out (line 4), and then it iteratively labels out any argument attacked by an argument labelled in (line 5). The iteration continues until no more arguments can be labelled in or out, and it terminates by labelling un any argument remained unlabelled (line 7).

Algorithm 1 Computation of a grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling

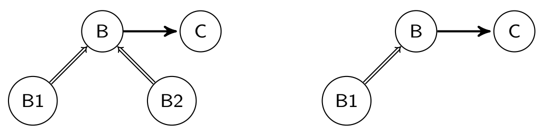
1: **input** Argumentation graph G,
2: $L_0 = (\emptyset, \emptyset, \emptyset)$,
3: **repeat**
4: $\quad \mathsf{in}(L_{i+1}) \leftarrow \mathsf{in}(L_i) \cup \{A | A \in \mathcal{A}_G$ is not labelled in L_i, and $\forall B \in \mathcal{A}_G$: if B attacks A then $B \in \mathsf{out}(L_i)\}$
5: $\quad \mathsf{out}(L_{i+1}) \leftarrow out(L_i) \cup \{A | A \in \mathcal{A}_G$ is not labelled in L_i, and $\exists B \in \mathcal{A}_G : B$ attacks A and $B \in \mathsf{in}(L_{i+1})\}$.
6: **until** $L_i = L_{i+1}$
7: **return** $(\mathsf{in}(L_i), \mathsf{out}(L_i), \mathcal{A}_G \backslash (\mathsf{in}(L_i) \cup \mathsf{out}(L_i)))$

This algorithm will constitute a basis for argumentative Boltzmann machines as we will see later, but for now we consider sub-graphs to prepare our probabilistic setting. We say that a sub-graph H of an argumentation graph G is *induced* if:

- for any argument A in $\mathcal{A}_H$, all the arguments supporting A are in $\mathcal{A}_H$,

- for any pair of arguments A and B in $\mathcal{A}_H$, $A \Rightarrow B$ is a support of H if, and only if, this support is a support of G.
- for any pair of arguments A and B in $\mathcal{A}_H$, $A \rightsquigarrow B$ is an attack of H if, and only if, this attack is an attack of G.

I.e. H is an induced sub-graph of G if any argument of H appears with all its supporting arguments and H has exactly the attacks and supports that appear in G over the same set of arguments (see Fig. 2).



(a) (b)

Figure 2: The graph (a) is a valid sub-graph of the graph in Fig. 1., while the graph (b) is not.

To re-concentrate on labellings, we now match any sub-graph with a legal $\{\mathsf{on}, \mathsf{off}\}$-labelling by "switching off" arguments outside the considered sub-graph, and we do the similar operation to define grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings.

Definition 5. [Legal $\{\mathsf{on}, \mathsf{off}\}$-labelling] Let H be a sub-graph of an argumentation graph G. A *legal* $\{\mathsf{on}, \mathsf{off}\}$-*labelling* of G with respect to H is a $\{\mathsf{on}, \mathsf{off}\}$-labelling of G such that

- every argument in $\mathcal{A}_H$ is labelled on,
- every argument in $\mathcal{A}_G \backslash \mathcal{A}_H$ is labelled off.

Definition 6. [Grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling] Let H be a sub-graph of an argumentation graph G. A *grounded* $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-*labelling* of G with respect to H is a $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling such that:

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

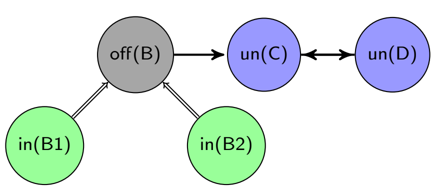


Figure 3: A grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling.

An argumentation graph G 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 induced by G.

As for notational matters, a legal $\{\mathsf{on}, \mathsf{off}\}$-labelling will be abbreviated $\{\mathsf{on}, \mathsf{off}\}^l$-labelling, a complete $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling 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 X-labellings of an argumentation graph G as $\mathcal{L}_G^X$, and each set will basically constitute a sample space of our probabilistic setting for argumentation with respect to a X-labelling.

Depending on the envisaged application, the probabilistic argumentation frame can be set in many ways. We propose a generic setting where the sample space can be the set of labellings of an argumentation graph called *a hypothetical*

B1	B2	B	C	D
on	on	on	on	on
on	on	on	on	off
on	on	on	off	on
on	on	on	off	off
on	on	off	on	on
on	on	off	on	off
on	on	off	off	on
on	on	off	off	off
on	off	off	on	on
on	off	off	on	off
on	off	off	off	on
on	off	off	off	off
off	on	off	on	on
off	on	off	on	off
off	on	off	off	on
off	on	off	off	off
off	off	off	on	on
off	off	off	on	off
off	off	off	off	on
off	off	off	off	off

B1	B2	B	C	D
in	in	in	out	in
in	in	in	out	off
in	in	in	off	in
in	in	in	off	off
in	in	off	un	un
in	in	off	in	off
in	in	off	off	in
in	in	off	off	off
in	off	off	un	un
in	off	off	in	off
in	off	off	off	in
in	off	off	off	off
off	in	off	un	un
off	in	off	in	off
off	in	off	off	in
off	in	off	off	off
off	off	off	un	un
off	off	off	in	off
off	off	off	off	in
off	off	off	off	off

Figure 4: Sample spaces of the hypothetical argumentation frame given in Fig. 1 with respect to the legal $\{\mathsf{on}, \mathsf{off}\}$-labelling (left) and the grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling (right). Each line represents a sample, i.e. a labelling.

argumentation frame with respect to any particular type of labelling. As a first step, and in order to relax any assumption on the probabilistic independence amongst arguments (c.f. [13, 15, 24, 4, 7]), any labelling is attached a potential, then the probabilistic frame is proposed.

Definition 7. [Potential of a labelling] Let G be an argumentation graph and let X indicates a type of labelling. A *potential of a X-labelling* $L \in \mathcal{L}_G^X$ is a function $Q : \mathcal{L}_G^X \to \mathbb{R}$.

Definition 8. [Probabilistic argumentation frame] A probabilistic argumentation frame is a tuple $(G, X, (\Omega, F, P))$ such that G is an argumentation graph (called the hypothetical argumentation frame), X indicates the type of the considered X-labelling and (Ω, F, P) is a probability space such that:

- the sample space Ω is the set of X-labellings w.r.t. G, $\Omega = \mathcal{L}_G^X$,
- the σ-algebra F is the power set of Ω,
- the probability function P from $F(\Omega)$ to $[0, 1]$ takes the form of a Gibbs-Boltzmann distribution:

$$P(\{L\}) = \frac{e^{-Q(L)}}{\sum_{\{L'\} \in F(\Omega)} e^{-Q(L')}} \tag{1}$$

where $Q(L)$ is the potential associated to any labelling L such that $\{L\} \in F(\Omega)$.

The proposed probabilistic setting is generic in the sense that it can host other types of labellings such as complete $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings, preferred $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings (not presented here) etc. We will focus on legal $\{\mathsf{on}, \mathsf{off}\}$-labellings and grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings.

Notice that a probabilistic argumentation frame where the labelling is a legal $\{\mathsf{on}, \mathsf{off}\}$-labelling boils down to the case where the sample space is the set of sub-graphs of the hypothetical argumentation frame, c.f. [13, 15, 24, 4, 7]. Moreover, since any legal $\{\mathsf{on}, \mathsf{off}\}$-labelling can be trivially mapped to one grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling and one only (as we can visualise in Fig. 4), a probabilistic argumentation frame with legal $\{\mathsf{on}, \mathsf{off}\}$-labellings is equivalent to one with grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings because they shall give the same probability results.

Whatever the selected type of labelling, the probability of the labelling of some arguments is the marginal probability over the samples where these arguments are labelled as such. Unfortunately, given a large hypothetical argumentation frame, the size of the sample space will explode. To tackle such explosion, we may consider approaches from graphical models (see [14]) to have a compact representation of the sample space, with associated techniques for learning and inference. And because we want to learn the dependencies amongst arguments (instead of fixing them from some background knowledge), we consider next a particular class of graphical models called Boltzmann machines before integrating these machines with probabilistic argumentation for learning and inference.

3. BOLTZMANN MACHINES

Boltzmann machines (BM) is a special case of Markov random fields with a possible interpretation as neural networks. These machines are a Monte-Carlo version of Hopfield networks inspired by models in statistical physics. A Boltzmann machine (BM) is an undirected graphical model. The nodes represent random variables whose values are either 1 or 0. Each node represents either a "visible" or a "hidden" unit, with the former modelling observations and the latter capturing the underlying pattern of dependencies between the visible nodes. An assignment of binary values to the visible or hidden variables will be denoted by $\mathbf{v} \in \{0,1\}^V$ and $\mathbf{h} \in \{0,1\}^H$. The edges represent pairwise symmetric interactions between the variables, each edge is associated with a weight. The weights between visible-to-hidden, visible-to-visible and hidden-to-hidden nodes are captured in matrices denoted $\mathbf{W}$, $\mathbf{L}$ and $\mathbf{J}$ respectively. The diagonales of $\mathbf{L}$ and $\mathbf{J}$ are set to 0. Let θ denote all weight parameters $\{\mathbf{W}, \mathbf{L}, \mathbf{J}\}$, a joint configuration $(\mathbf{v}, \mathbf{h}, \theta)$ of the visible and hidden nodes has an energy given by:[1]

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\mathbf{v}^\top \mathbf{L} \mathbf{v} - \mathbf{h}^\top \mathbf{J} \mathbf{h} - \mathbf{v}^\top \mathbf{W} \mathbf{h} \tag{2}$$

The energy of a configuration defines the probability of this configuration via a Gibbs-Boltzmann distribution:

$$P(\mathbf{v}, \mathbf{h}; \theta) = \frac{e^{-E(\mathbf{v}, \mathbf{h}; \theta)}}{\sum_{\mathbf{v}', \mathbf{h}'} e^{-E(\mathbf{v}', \mathbf{h}'; \theta)}} \tag{3}$$

The energy of activation of a node is the sum of the weights of connections from other active nodes:

$$Q(v_i = 1) = \sum_{j=1}^{H} W_{ij} h_j + \sum_{k=1 \setminus i}^{V} L_{ik} v_i \tag{4}$$

$$Q(h_j = 1) = \sum_{i=1}^{V} W_{ij} v_i + \sum_{k=1 \setminus j}^{H} J_{jk} h_j \tag{5}$$

In the remainder, we may abbreviate $Q(v_i = 1)$ as $Q(v_i)$. The conditional probabilities of activation for the hidden and visible nodes are as follows:

$$P(v_i = 1 | \mathbf{h}, \mathbf{v}_{-i}; \theta) = \sigma(Q(v_i = 1)) \tag{6}$$

[1]Here we omit biases for the sake of clarity.

$$P(h_j = 1 | \mathbf{v}, \mathbf{h}_{-j}; \theta) = \sigma(Q(h_j = 1)) \tag{7}$$

where $\sigma(x) = 1/\left(1 + e^{-x}\right)$ is the sigmoid logistic function.

When the machine is running "freely", weights are fixed and configurations are sampled using the above conditional probabilities. When a machine is learning, weights and values of hidden variables are adjusted so that the probability distribution of the visible variables shall fit the distribution of a training dataset. A measure of the discrepancy between the distribution of a training dataset, denoted $P(\mathbf{v})$, and the distribution of produced configurations of visible nodes, denoted $P'(\mathbf{v})$, is the Kullback-Leibler divergence D_{KL}:

$$D_{KL}(P \parallel P') = \sum_{\mathbf{v}} P(\mathbf{v}) \ln \frac{P(\mathbf{v})}{P'(\mathbf{v})} \tag{8}$$

To minimize the divergence, Hinton and Sejnowski [12] proposed to change the weights proportionally to the difference between the average probability of two units being switched on when the visibles nodes are clamped with a training example, denoted P_{ij}, and the corresponding probability P'_{ij} when the machine is in free mode:

$$\Delta w_{ij} = \epsilon (P_{ij} - P'_{ij}) \tag{9}$$

where ϵ scales the change. Constraints can be set on weights to improve learning, for example, when $\mathbf{J} = 0$ and $\mathbf{L} = 0$ then we obtain restricted Boltzmann machines (RBM), leading eventually to deep Boltzmann machines [22]. Thus, a restricted Boltzmann machine consists of a layer of hidden nodes and a layer of visible nodes with no visible-visible or hidden-hidden connections. With these restrictions, the visible (hidden) nodes are conditionally independent given a hidden (visible) vector, and samples for P_{ij} and P'_{ij} can be consequently computed with alternating blocked Gibbs sampling. On this basis, an efficient learning procedure called contrastive divergence (CD-n) was proposed to approximate gradient descent, with good performance in practice [10].

Once a machine is trained, it provides a compact representation of the distribution of the observations. A machine is then possibly used in two modes: the generative mode and the discriminative mode. In the generative mode, the machine is run in free mode without clamping any visible node for randomly generating data as it has been learned. In the discriminative mode, the machine will produce data only for some variable(s) conditional on fixed variables. In practice, the visible nodes corresponding to these fixed variables are clamped and the machine is then run in free mode to sample the remaining visible units. The discriminative mode is typically used for classification tasks or to complete partial observations.

4. ARGUMENTATIVE MACHINES

Different machines can be set up to embody probabilistic argumentation and we are in fact in front of a class of Boltzmann machines, that we call the class of argumentative Boltzmann machines, in the sense that different machines can be proposed to label arguments. In this paper, we reformulate labelling as a binary problem, we propose two argumentative Boltzmann machines, and we will focus on the second.

A simple solution consists in a machine where every argument is represented by a visible node. We call this machine a $\{\mathsf{on}, \mathsf{off}\}$-labelling Boltzmann machine. In this machine, an argument is on (thus labelled either in or out or un with respect to the grounded semantics) if, and only if, the corresponding node is switched on, and an argument is off (i.e. labelled either off) if, and only if, the corresponding node is switched off. When learning, any example clamped to the machine is a $\{\mathsf{on}, \mathsf{off}\}$-labelling (i.e. a sub-graph) to indicate whether an argument is on or off. To ensure the production of legal $\{\mathsf{on}, \mathsf{off}\}$-labellings, an argument shall be labelled on on the necessary (but not sufficient) condition that all the nodes of its supporting arguments are switched on. So, a legal $\{\mathsf{on}, \mathsf{off}\}$-labelling machine shall learn a distribution of $\{\mathsf{on}, \mathsf{off}\}$-labellings and, when running in free mode, it shall sample legal $\{\mathsf{on}, \mathsf{off}\}$-labellings according to the learned distribution. Once a legal $\{\mathsf{on}, \mathsf{off}\}$-labelling is given, one can compute the entailed grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling using Alg. 1. The advantage of such a machine is its simplicity: it boils down to a conventional Boltzmann machine chained with an algorithm to compute the grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling of sampled graphs (or any other labelling based on the provision of such algorithm to compute such labelling). A major disadvantage is that in, out or un labellings cannot be discriminated (limiting the discriminative mode). Finally one may argue that this solution is a shallow or boxy construction for neuro-argumentative agents.

Towards deeper integration between argumentation and Boltzmann machines, and to address the limitations of legal $\{\mathsf{on}, \mathsf{off}\}$-labelling Boltzmann machines, we investigate a variant where a configuration of visible nodes represents a $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling. Considering a hypothetical argumentation frame, we associate a Boltzmann machine, henceforth referred to as a *grounded* $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$*-labelling Boltzmann machine*, for which every label in, out, un and off of every argument is paired with one visible node, and only one. When a visible node has value 1 then the argument shall be labelled accordingly, if the node has value 0 then there is no such labelling. As for notation, the visible node representing the argument A labelled in will be denoted $v_{\mathsf{in}(A)}$, and similarly $v_{\mathsf{out}(A)}$ will denote A labelled out, $v_{\mathsf{un}(A)}$ for A labelled un, and $v_{\mathsf{off}(A)}$ for A labelled off. An example of an argumentative Boltzmann machine is given in Fig. 5.

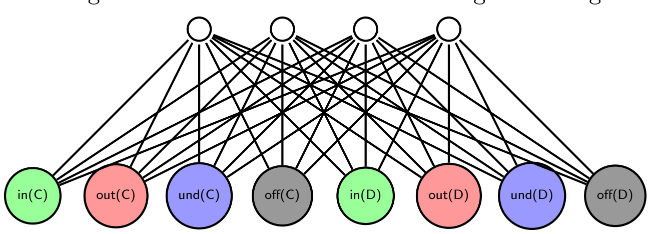


Figure 5: A grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling restricted Boltzmann machine with two arguments C and D. Each visible node corresponds to a label of an argument.

Accordingly, a grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling L corresponds to a configuration $\mathbf{v}_L$, and the probability of this labelling is the marginal probability of this configuration:

$$P(\mathbf{v}_L; \theta) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v}_L, \mathbf{h}; \theta)}}{\sum_{\mathbf{v'}, \mathbf{h'}} e^{-E(\mathbf{v'}, \mathbf{h'}; \theta)}} \tag{10}$$

We can marginalise over the hidden nodes and define $P(\mathbf{v}_L; \theta)$ in terms of the "free energy" $F(\mathbf{v}_L)$ to obtain:

$$P(\mathbf{v}_L; \theta) = \frac{e^{-F(\mathbf{v}_L)}}{\sum_{\mathbf{v'}} e^{-F(\mathbf{v'})}} \tag{11}$$

In this view, the potentials of labellings (see Def. 7) is meant to be approximated by the corresponding free energies (up to a constant). Notice that the free energy of visible nodes takes a time linear in the number of hidden units, see [11], and thus any ratio of probabilities between two labellings L and L' can be computed in linear time as follows:

$$\frac{P(\mathbf{v}_L; \theta)}{P(\mathbf{v}_{L'}; \theta)} = e^{-F(\mathbf{v}_L) + F(\mathbf{v}_{L'})} \tag{12}$$

When sampling arguments, we may check ex post whether a produced labelling is grounded and discard it if not, but this solution may involve extra computation that shall slow down learning and inferences. So we propose to turn the algorithm of grounded labellings (Alg. 1) into a random walk (Alg. 2) to ensure the production of grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings.

For the sake of clarity, we adopt the following notation. Let G denote a hypothetical argumentation frame. An argument A is labelled in with respect to a labelling L_i (we are indexing the labelling as they will appear in Alg. 2) if

- A is not labelled in L_i, and
- any supporting argument is in, i.e. $\forall A' \in \mathcal{A}_G : A'$ supports A, $A' \in \mathsf{in}(L_i)$, and
- any attacking argument is either out or off, i.e. $\forall B \in \mathcal{A}_G : B$ attacks A, $B \in \mathsf{out}(L_i) \cup \mathsf{off}(L_i)$,
- A is drawn in: $u \leq e^{Q(v_{\mathsf{in}(A)})}/Z(A)$ where u is a random number in [0,1] drawn from an uniform distribution, and $Z(A) = e^{Q(v_{\mathsf{in}(A)})} + e^{Q(v_{\mathsf{out}(A)})} + e^{Q(v_{\mathsf{un}(A)})} + e^{Q(v_{\mathsf{off}(A)})}$

We denote $\mathsf{IN}(L_i)$ the set of arguments eventually labelled in with respect to a labelling L_i. An argument A fulfilling the first three conditions but not drawn in is said inable, abbreviated $\mathsf{inable}(L_i, A)$.

An argument A is labelled out with respect to labellings L_i and L_{i+1} if

- A is not labelled in L_i, and
- any supporting argument is labelled in or out or un, i.e. $\forall A' \in \mathcal{A}_G : A'$ supports A, $A' \in \mathsf{un}(L_i) \cup \mathsf{out}(L_i) \cup \mathsf{in}(L_{i+1})$, and
- there exists an attacking argument labelled in, i.e. $\exists B \in \mathcal{A}_G : B$ attacks A and $B \in \mathsf{in}(L_{i+1})$.
- A is drawn out: $u \leq e^{Q(v_{\mathsf{out}(A)})}/Z(A)$.

We denote $\mathsf{OUT}(L_i, L_{i+1})$ the set of arguments eventually labelled out with respect to the labellings L_i and L_{i+1}. An argument A fulfilling the first three conditions but not drawn out is said outable, abbreviated $\mathsf{outable}(L_i, L_{i+1}, A)$.

An argument A is labelled off with respect to labellings L_i and L_{i+1} if

- A is not labelled in L_i, and
- A was not labelled in or out, i.e. $\mathsf{inable}(L_i, A)$ or $\mathsf{outable}(L_i, L_{i+1}, A)$.

We denote $\mathsf{OFF1}(L_i, L_{i+1})$ the set of arguments labelled off this way. Furthermore, an argument A is also labelled off with respect to labellings L and L_{i+1} if

- A is not labelled in L, and
- A is supported by an argument B which is labelled off in L_{i+1}.

We denote $\mathsf{OFF2}(L, L_{i+1})$ the set of arguments which are labelled off for this reason. Finally, an argument A is labelled off with respect to a labelling L_{i+1} if

- A is not labelled in L_{i+1}, and
- A is drawn off: $u \leq e^{Q(v_{\mathsf{off}(A)})}/Z(A)$.

We denote $\mathsf{OFF3}(L_{i+1})$ the set of arguments labelled off this way.

We can now present a simple version of the grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling random walk as given in Alg. 2.

Algorithm 2 $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}^g$-labelling random walk.

1: **input** Machine parameters θ, argumentation graph G,
2: $L_0 = (\emptyset, \emptyset, \emptyset, \emptyset)$,
3: **repeat**
4: $\quad L'_0 \leftarrow L_i$
5: $\quad$ **repeat**
6: $\quad\quad \mathsf{in}(L'_{i+1}) \leftarrow \mathsf{in}(L'_i) \cup \mathsf{IN}(L'_i)$
7: $\quad\quad \mathsf{out}(L'_{i+1}) \leftarrow \mathsf{out}(L'_i) \cup \mathsf{OUT}(L'_i, L'_{i+1})$
8: $\quad\quad \mathsf{off}(L'_{i+1}) \leftarrow \mathsf{off}(L'_i) \cup \mathsf{OFF1}(L'_i, L'_{i+1})$
9: $\quad\quad \mathsf{off}(L'_{i+1}) \leftarrow \mathsf{off}(L'_{i+1}) \cup \mathsf{OFF2}(L'_i, L'_{i+1})$
10: $\quad$ **until** $L'_i = L'_{i+1}$
11: $\quad L_{i+1} \leftarrow L'_{i+1}$
12: $\quad \mathsf{off}(L_{i+1}) \leftarrow \mathsf{off}(L_{i+1}) \cup \mathsf{OFF3}(L_{i+1})$.
13: $\quad \mathsf{off}(L_{i+1}) \leftarrow \mathsf{off}(L_{i+1}) \cup \mathsf{OFF2}(L_{i+1}, L_{i+1})$.
14: **until** $L_i = L_{i+1}$
15: **return** the $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling:
$(\mathsf{in}(L_i), \mathsf{out}(L_i), \mathcal{A}_G - \mathsf{in}(L_i) \cup \mathsf{out}(L_i) \cup \mathsf{off}(L_i), \mathsf{off}(L_i))$

A sampling of visible nodes (i.e. labels of arguments) performed with the $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}^g$-labelling random walk (also abbreviated grounded labelling walk in the remainder) is called a grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-sampling or simply a grounded labelling sampling.

The walk consists of an outer loop and an inner loop. The inner loop draws the labelling in and out of arguments with respect to their probability learned by the machine. The inner loop begins considering all arguments not being attacked and potentially label each of these arguments as in (line 6). Then the algorithm proceeds by considering any argument attacked by an argument labelled in, and potentially label each of these arguments as out (line 7). If an argument eligible for a label in or out is not labelled as such, then it is labelled off (line 8), and unsupported arguments get also labelled off (line 9). The iteration of the inner loop continues until no more arguments can be further labelled (line 10). Any supported argument remaining unlabelled is then labelled off with its respective probability (line 12), and any new unsupported argument is labelled off (line 13). If there is an argument labelled off, then the labelling returns in the inner loop to potentially label some arguments in, out or off. If there is no argument labelled off, then the walk terminates by eventually labelling un all remaining arguments (line 15).

When learning, the machine will be shown a set of training examples where an example is a labelling. To clamp the visible nodes of a machine with a labelling, a visible node will be set at 1 if its corresponding argument is labelled as such, otherwise it is set at 0. If a training example does not match any grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling of the hypothetical framework then it suggests that the hypothetical framework should be changed and be adapted, but noisy settings may also occur. We reserve these issues for future investigations.

Notice that, the order in which the arguments are labelled matters in the case of a conventional BM while the order of labelling does not matter in the case of a RBM. Thus an argumentative machine based on a conventional BM exhibits a discrepancy between an ideal purely random walk amongst possible states of the machine and the proposed grounded labelling walk. Indeed, this labelling walk is not purely random, as visible nodes are now considered in an order regimented by the construction of a grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling. To integrate the labelling constraints, we will thus favour RBM so that the sampling of the visible nodes can be performed through the grounded labelling walk.

Example 1. Let's illustrate the grounded labelling walk with the frame in Fig. 1. We may have the following walk:

1. $L_0 = (\emptyset, \emptyset, \emptyset, \emptyset)$,
   1.1 $L'_0 = (\emptyset, \emptyset, \emptyset, \emptyset)$,
      1.1.1 say $u < e^{Q(v_{\mathsf{in}(\mathsf{B}_1)})}/Z(\mathsf{B}_1)$,
      and $u < e^{Q(v_{\mathsf{in}(\mathsf{B}_2)})}/Z(\mathsf{B}_2)$,
      thus: $\mathsf{in}(L'_1) = \{\mathsf{B}_1, \mathsf{B}_2\}$.
   1.2 $L'_1 = (\{\mathsf{B}_1, \mathsf{B}_2\}, \emptyset, \emptyset, \emptyset)$
      1.2.1 say $u < e^{Q(v_{\mathsf{in}(\mathsf{B})})}/Z(\mathsf{B})$,
      thus $\mathsf{in}(L'_2) = \{\mathsf{B}_1, \mathsf{B}_2, \mathsf{B}\}$,
      1.2.2 say $u > e^{Q(v_{\mathsf{out}(\mathsf{C})})}/Z(\mathsf{C})$,
      thus: $\mathsf{out}(L'_2) = \emptyset$.
      1.2.3 $\mathsf{off}(L'_2) = \{\mathsf{C}\}$.
   1.3 $L'_2 = (\{\mathsf{B}_1, \mathsf{B}_2, \mathsf{B}\}, \emptyset, \emptyset, \{\mathsf{C}\})$.
      1.3.1 say $u < e^{Q(v_{\mathsf{in}(\mathsf{D})})}/Z(\mathsf{D})$,
      thus $\mathsf{in}(L'_3) = \{\mathsf{B}_1, \mathsf{B}_2, \mathsf{B}, \mathsf{D}\}$.
      1.3.2 $\mathsf{out}(L'_3) = \emptyset$
      1.3.3 $\mathsf{off}(L'_3) = \{\mathsf{C}\}$
   1.4 $L'_3 = (\{\mathsf{B}_1, \mathsf{B}_2, \mathsf{B}, \mathsf{D}\}, \emptyset, \emptyset, \{\mathsf{C}\})$
   1.5 $L'_4 = L'_3$.
2. $L_1 = (\{\mathsf{B}_1, \mathsf{B}_2, \mathsf{B}, \mathsf{D}\}, \emptyset, \emptyset, \{\mathsf{C}\})$.
3. $L_2 = L_1$.

The resulting labelling is $(\{\mathsf{B}_1, \mathsf{B}_2, \mathsf{B}, \mathsf{D}\}, \emptyset, \emptyset, \{\mathsf{C}\})$. Another walk may be:

1. $L_0 = (\emptyset, \emptyset, \emptyset, \emptyset)$,
   1.1 $L'_0 = (\emptyset, \emptyset, \emptyset, \emptyset)$,
      1.1.1 say $u < e^{Q(v_{\mathsf{in}(\mathsf{B}_1)})}/Z(\mathsf{B}_1)$,
      and $u > e^{Q(v_{\mathsf{in}(\mathsf{B}_2)})}/Z(\mathsf{B}_2)$,
      thus: $\mathsf{in}(L'_1) = \{\mathsf{B}_1\}$.
      1.1.3 $\mathsf{off}(L'_1) = \{\mathsf{B}_2, \mathsf{B}\}$.
   1.2 $L'_1 = (\{\mathsf{B}_1\}, \emptyset, \emptyset, \{\mathsf{B}_2, \mathsf{B}\})$
   1.3 $L'_1 = L'_2$
2. say $u > e^{Q(v_{\mathsf{off}(\mathsf{C})})}/Z(\mathsf{C})$, and $u > e^{Q(v_{\mathsf{off}(\mathsf{D})})}/Z(\mathsf{D})$,
   thus $L_1 = (\{\mathsf{B}_1\}, \emptyset, \emptyset, \{\mathsf{B}_2, \mathsf{B}\})$.
3. $L_2 = L_1$

So this walk returns the labelling $(\{\mathsf{B}_1\}, \emptyset, \{\mathsf{C}, \mathsf{D}\}, \{\mathsf{B}_2, \mathsf{B}\})$.

Theorem 1. [Termination] The $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}^g$-labelling random walk (Alg. 2) terminates.

Proof. We consider the sequence of pairs $(A_0, L_0), (A_1, L_1), \ldots, (A_n, L_n)$ where A_i and L_i are the set of unlabelled arguments and the set of labelled arguments (respectively) at the beginning of the iteration i of the outer loop. At each iteration i, the cardinality of A_i is strictly inferior to the cardinality of A_{i-1}. At some point, there is no further argument to label, then $L_n = L_{n-1}$ and thus the algorithm exits. □

Theorem 2. [Soundness] The $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}^g$-labelling random walk is sound.

Proof. We consider any terminated sequence $(A_0, L_0), (A_1, L_1), \ldots, (A_n, L_n)$, and we show that L_n is a grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling. We consider the labelled argumentation sub-graph H induced by the arguments not labelled off and we observe that this graph has a grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}\}$-labelling L' such that this labelling L' is complete, and $\mathsf{in}(L')$ is minimal (because no less arguments can be labelled in). Thus L_n is a grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling. □

Theorem 3. [Completeness] The $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}^g$-labelling random walk is complete.

PROOF. We demonstrate that for any grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling L, there exists a terminated sequence $(A_0, L_0), (A_1, L_1), \ldots, (A_n, L_n)$ where $L_n = L$. For any terminated sequence L, there exists a unique graph induced by the arguments labelled in, out or un in L, and thus we have to show that the walk can return this graph. However, the walk can return any set of arguments labelled off. Consequently, for any grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling L, there exists a terminated sequence $(A_0, L_0), (A_1, L_1), \ldots, (A_n, L_n)$ with $L_n = L$. □

Theorem 4. [Time complexity] Time complexity of the $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}^g$-labelling random walk is polynomial.

PROOF. At each iteration of the inner loop, one argument at least is labelled, otherwise the algorithm terminates. Therefore, when the input argumentation frame has N arguments, then there are N iterations. For each iteration, the status of N argument has to be checked with respect to the status of N arguments in the worst case. The time complexity of this inner loop is thus polynomial. When the inner loop terminates, one iteration of the outer loop completes by checking the status of remaining unlabelled arguments, and the outer loop iterates N times at worst. Therefore, the time complexity of the walk is polynomial. □

Theorem 5. [Space complexity] Space complexity of the $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}^g$-labelling random walk is $\mathrm{O}(max(|\mathbf{v}| \times |\mathbf{v}|, |\mathbf{h}| \times |\mathbf{h}|, |\mathbf{v}| \times |\mathbf{h}|))$.

PROOF. A Boltzmann machine can be described by two binary random vectors corresponding to nodes vectors $\mathbf{v}$ and $\mathbf{h}$, and matrices $\mathbf{L}$, $\mathbf{J}$ and $\mathbf{W}$ (corresponding to the strength of the edges between hidden and visible nodes, see Figure 5), an hypothetical argumentation graph and machine parameters. The two binary vectors, the hypothetical argumentation graph and machine parameters require minimal space, thus the memory requirements are dominated by the real-valued edge matrices $\mathbf{L}$, $\mathbf{J}$ and $\mathbf{W}$. We need a weight to describe each pairwise interaction amongst hidden and visible nodes, thus it holds that the dimensions of the matrices are $|\mathbf{v}| \times |\mathbf{v}|$ for $\mathbf{L}$, $|\mathbf{h}| \times |\mathbf{h}|$ for $\mathbf{J}$, and $|\mathbf{v}| \times |\mathbf{h}|$ for $\mathbf{W}$. □

In case of a restricted Boltzmann machine, we have $\mathbf{L} = 0$ and $\mathbf{J} = 0$, therefore the space complexity is reduced to $\mathrm{O}(|\mathbf{v}| \cdot |\mathbf{h}|)$, which greatly contrasts with the space complexity of the sample space of our probabilistic setting.

As to the practical use of the walk, we may alternate grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-samplings and ordinary block samplings with a mixing rate $p_m \in [0, 1]$, so that the grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-sampling replaces the ordinary block sampling of any RBM with a probability p_m. Hence, if the rate $p_m = 0$ then only grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling samplings are performed. If the rate $p_m = 1$, then we have a conventional RBM. The effect of different rates will be experimentally evaluated in Section 5.

An argumentative Boltzmann machine can be used in a generative or a discriminative mode. In both modes, argumentation is meant to constrain the sample space. In this paper, we focus on a pure argumentative sample space in the sense that any sample is a labelling, but we can extend it to build multi-modal machines where some visible nodes are the elements of observations (e.g., one visible unit for each pixel of a digital image). Notice that a visible node for the element of an observation can be understood as an argument with no explicit support or attack from other arguments. As the relations of support or attack of the argumentative framework are meant to be a prior knowledge, they may be particularly useful for some sort of argumentative classification. However, the proposed grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling random walk in the discriminative mode can only be used when fixing arguments to in or off. The generative mode does not suffer this limitation though. A labelling walk allowing a discriminative mode conditioned to out or un labels is left for future research.

5. EXPERIMENTS

Our experiments concern a comparison of trainings of a conventional Boltzmann machine and grounded $\{\mathsf{in}, \mathsf{out}\ \mathsf{un}, \mathsf{off}\}$-labelling machines.

The training of each machine was tested with 50 different datasets, each dataset was artificially generated (without any noise, i.e. all labellings are grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labellings) with respect to a common hypothetical argumentative frame shown in Fig. 6 - inducing $4^{14} = 268,435,456$ different possible $\{\mathsf{in}, \mathsf{out}\ \mathsf{un}, \mathsf{off}\}$-labellings, amongst which 3456 grounded labellings. Each dataset had a different entropy, ranging from 0 to $\ln(3456) \approx 8.14$ (the maximum).

Every training consisted of 3000 passes[2] over small mini-batches, each containing 100 labellings. The weights were updated after each mini-batch, using a learning rate from 0.6 at the beginning of the training to 0.1 at the end, momentum of 0.9, and a weight decay of 0.003 (see [11]). Every machine had 56 hidden nodes (the number of arguments' labels). The learning procedure was CD-1 (see [10]).

The statistical distance between the training distribution P and the distribution P' of sampled labellings produced during training by a machine was measured using the Kullback-Leibler divergence and the total variation distance:

$$\delta(P, P') = \frac{1}{2} \sum_{L \in \mathcal{L}_G^X} |P(L) - P'(L)| \tag{13}$$

The resulting distances with respect to the different datasets are given in Fig. 7.[3]

Our control is a conventional RBM (i.e. a labelling machine with mixing rate $p_m = 1$) chained with a module discarding produced $\{\mathsf{in}, \mathsf{out}\ \mathsf{un}, \mathsf{off}\}$-labellings which are not grounded - see Fig. 7 top. Of course longer training periods and better learning settings may give better results.

A "pure" grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling machine with a mixing rate $p_m = 0$ was then tested, see Fig. 7. With respect to the conventional RBM, this pure labelling machine produced distributions with higher distances for large entropies.

[2]Around 20 min. training for a non-optimised SWI-Prolog implementation (!), with an Intel i7-3770 CPU, 3.90 GHz.

[3]The Kullback-Leibler divergence is not drawn when not defined (as a labelling was not produced during a training).

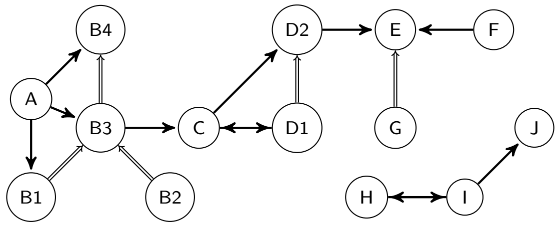


Figure 6: The hypothetical argumentation frame used for our experimental evaluation.

A grounded $\{\mathsf{in}, \mathsf{out}\ \mathsf{un}, \mathsf{off}\}$-labelling machine with a mixing rate $p_m = 0.5$ outperformed the conventional RBM, and resulting distances are less spread than those of the conventional RBM. This experiment suggests it is preferable to have a grounded $\{\mathsf{in}, \mathsf{out}\ \mathsf{un}, \mathsf{off}\}$-labelling machine with a mixing rate balancing conventional samplings and grounded $\{\mathsf{in}, \mathsf{out}\ \mathsf{un}, \mathsf{off}\}$-samplings.

The results reveal thus that a pure argumentative learner shall not train as well than a conventional learner, while a well balanced learner shall produce distributions with smaller statistical distances with respect to a conventional learner. In fact, the main advantage of the grounded labelling walk is the production of valid grounded $\{\mathsf{in}, \mathsf{out}\ \mathsf{un}, \mathsf{off}\}$-labellings *on demand*, so that some computational time is saved by immediately discarding labellings not produced by a grounded $\{\mathsf{in}, \mathsf{out}\ \mathsf{un}, \mathsf{off}\}$-sampling.

6. CONCLUSION

Towards neuro-argumentative agents, we proposed an integration of probabilistic abstract argumentation (accounting for sub-arguments) and Boltzmann machines for learning and labelling. The proposed probabilistic setting relaxes any assumption of probabilistic independences amongst arguments, it can accommodate different types of labelling, and we show how it can be seamlessly integrated to Boltzmann machines. We proposed two types of argumentative machines: the $\{\mathsf{on}, \mathsf{off}\}$-labelling machines and the grounded $\{\mathsf{in}, \mathsf{out}, \mathsf{un}, \mathsf{off}\}$-labelling machines, we focused on the second. We proved the termination, soundness and completeness of such machines along tractable computational complexity. Experiments revealed that a pure argumentative learner shall not train as well than a conventional learner, while a well balanced learner shall produce distributions with smaller statistical distances with respect to a conventional learner. The main advantage of the grounded labelling walk is the production of grounded labellings on demand, thereby saving computational time.

The combination of argumentation and neural networks is not new. D'Avila Garcez et al. proposed in [3] a neural model of argumentation with suggestions of applications in the legal domain. We share a connectivist approach for argumentation, but D'Avila Garcez et al. do not cater for any principled probabilistic setting. They selected directed graphs for their connectivist model of argumentation, while we opted for undirected graph model akin to Markov random fields to better account for our probabilistic setting.

Future developments range from instantiations of the abstract argumentation into more specific frameworks to the investigation of different types of labellings. We can extend a machine to build multi-modal machines where some visible nodes are the elements of observations (e.g., rhetoric elements). To complete the epistemic mechanism, one may endow agents with structure learning so that these agents shall learn the structure of a hypothetical frame, i.e. the relations of attack and support amongst arguments. A practical apparatus of neuro-argumentative agents towards action could be developed by integrating reinforcement learning and Boltzmann machines (see e.g. [23]) along argumentation. Argumentative machines may be used in argumentative multi-agent simulations (such as[21]) to account for stochastic behaviours, where argumentation shall provide scrutiny on the simulated behaviours. In game-theoretical argumentative settings [20], neuro-argumentative agents shall maintain a probability distribution over the set state features to make better decisions. However, argumentation frameworks may not be fixed, arguments and attacks may be introduced incrementally. In such dynamic settings, a brutal approach is to retrain a machine from scratch - but more subtile approaches are left for future research.

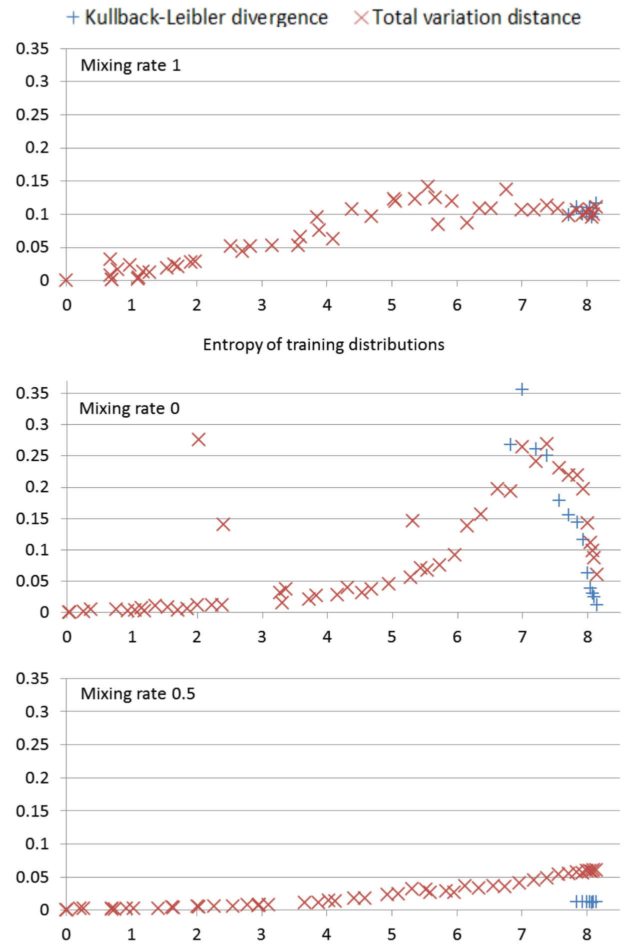


Figure 7: Statistical distances versus the entropy of training distributions.

Acknowledgement

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

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

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

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

[6] P. M. Dung and P. M. Thang. Towards (probabilistic) argumentation for jury-based dispute resolution. In *Proceedings of the 3rd International Conference on Computational Models of Argument*, pages 171–182. IOS Press, 2010.

[7] B. Fazzinga, S. Flesca, and F. Parisi. On the complexity of probabilistic abstract argumentation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 898–904. AAAI Press, 2013.

[8] A. S. d. Garcez, L. C. Lamb, and D. M. Gabbay. *Neural-Symbolic Cognitive Reasoning*. Springer, 1 edition, 2008.

[9] G. Governatori, M. J. Maher, G. Antoniou, and D. Billington. Argumentation semantics for defeasible logic. *J. Log. Comput.*, 14(5):675–702, 2004.

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

[11] G. E. Hinton. A practical guide to training restricted boltzmann machines. In *Neural Networks: Tricks of the Trade - Second Edition*, volume 7700, pages 599–619. Springer, 2012.

[12] G. E. Hinton and T. J. Sejnowski. Optimal perceptual inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1983.

[13] A. Hunter. A probabilistic approach to modelling uncertain logical arguments. *International Journal of Approximating Reasoning*, 54(1):47–81, 2013.

[14] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[15] H. Li, N. Oren, and T. J. Norman. Probabilistic argumentation frameworks. In *Revised Selected Papers of First International Workshop Theory and Applications of Formal Argumentation*, pages 1–16. Springer, 2011.

[16] S. Modgil and M. Caminada. Proof theories and algorithms for abstract argumentation frameworks. In *Argumentation in artificial intelligence*, pages 105–129. Springer, 2009.

[17] S. Modgil and H. Prakken. The $ASPIC^{+}$ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.

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

[19] H. Prakken. On support relations in abstract argumentation as abstractions of inferential relations. In *Proceedings of the 21st European Conference on Artificial Intelligence*, pages 735–740. IOS Press, 2014.

[20] R. Riveret, H. Prakken, A. Rotolo, and G. Sartor. Heuristics in argumentation: A game theory investigation. In *Proceedings of the 2nd International Conference on Computational Models of Argument*, pages 324–335. IOS Press, 2008.

[21] R. Riveret, A. Rotolo, and G. Sartor. Probabilistic rule-based argumentation for norm-governed learning agents. *Artificial Intelligence and Law*, 20(4):383–420, 2012.

[22] R. Salakhutdinov and G. E. Hinton. Deep Boltzmann Machines. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, volume 5 of *JMLR Proceedings*, pages 448–455. JMLR.org, 2009.

[23] B. Sallans and G. E. Hinton. Reinforcement learning with factored states and actions. *Journal of Machine Learning Research*, 5:1063–1088, 2004.

[24] M. Thimm. A probabilistic semantics for abstract argumentation. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 750–755. IOS Press, 2012.