

## Probabilistic abstract argumentation: an investigation with Boltzmann machines

Régis Riveret<sup>a\*</sup>, Dimitrios Korkinof<sup>b</sup>, Moez Draief<sup>c</sup> and Jeremy Pitt<sup>d</sup>

<sup>a</sup>Data61, Brisbane, Australia; <sup>b</sup>Cortexica Vision Systems, London, UK; <sup>c</sup>Huawei Technologies, Paris, France; <sup>d</sup>Imperial College, London, UK

(Received 24 June 2015; accepted 8 October 2015)

Probabilistic argumentation and neuro-argumentative systems offer new computational perspectives for the theory and applications of argumentation, but their principled construction involves two entangled problems. On the one hand, probabilistic argumentation aims at combining the quantitative uncertainty addressed by probability theory with the qualitative uncertainty of argumentation, but probabilistic dependences amongst arguments as well as learning are usually neglected. On the other hand, neuro-argumentative systems offer the opportunity to couple the computational advantages of learning and massive parallel computation from neural networks with argumentative reasoning and explanatory abilities, but the relation of probabilistic argumentation frameworks with these systems has been ignored so far. Towards the construction of neuro-argumentative systems based on probabilistic argumentation, we associate a model of abstract argumentation and the graphical model of Boltzmann machines (BMs) in order to (i) account for probabilistic abstract argumentation with possible and unknown probabilistic dependences amongst arguments, (ii) learn distributions of labellings from a set of cases and (iii) sample labellings according to the learned distribution. Experiments on domain independent artificial datasets show that argumentative BMs can be trained with conventional training procedures and compare well with conventional machines for generating labellings of arguments, with the assurance of generating grounded labellings – on demand.

**Keywords:** machine learning; probabilistic argumentation; graphical models; Boltzmann machines; neuro-argumentative systems; product of experts

### 1. Introduction

The combination of formal argumentation and probability theory to obtain formalisms catering for qualitative uncertainty as well as qualitative uncertainty has attracted much attention in the recent years, see, for example, Li et al. (2011), Thimm (2012), Hunter (2013), Sneyers et al. (2013) and Dondio (2014) and applications have been suggested, for example, in the legal domain by Roth et al. (2007), Riveret et al. (2007), Riveret et al. (2008) and Dung and Thang (2010). The underlying argumentation frameworks, the probability interpretation, the probability spaces and the manipulation of probabilities vary due to the different interests (see next section for details), but settings dealing with probabilistically dependent arguments, along issues regarding the computational complexity of inferences and learning, have been hardly explored so far.

However, the pursuit for efficient inference and learning in domains possibly structured by some logic-based knowledge is not new. For instance, statistical relational learning (SRL) (Getoor and Taskar, 2007) is a burgeoning field at the crossroad of statistics, probability theory, logics

---

\*Corresponding author. Email: [regis.riveret@nicta.com.au](mailto:regis.riveret@nicta.com.au)

and machine learning, and a plethora of frameworks have been proposed, see, for example, probabilistic relational models (Getoor et al., 2007) and Markov logic networks (Richardson and Domingos, 2006). Inspired by these approaches consisting in the combination of logic-based systems and graphical models to handle probabilistic dependences, we are willing to investigate the use of graphical models to tackle dependences in a setting of probabilistic abstract argumentation.

Another interesting and parallel active line of research possibly combining probability theory, logics and machine learning regards neuro-symbolic systems meant to join the strength of neural network models and logics (d’Avila Garcez et al., 2008): neural networks offer sturdy learning with the possibility of massive parallel computation, while logic brings intelligible knowledge representation and reasoning with explanatory ability, thereby facilitating the transmission of learned knowledge to other agents. Considering the importance of defeasible reasoning for agents reasoning with incomplete information and the intuitive account of this type of reasoning by formal argumentation, argumentation shall provide an ergonomic representation and argumentative reasoning aptitude to the underlying neural networks. However, such neuro-argumentation has been hardly explored so far, cf. d’Avila Garcez et al. (2014), and thus we are willing to investigate a possible construction of neuro-argumentative systems with a principled account of probabilistic argumentation.

*Contribution.* We investigate an integration of abstract argumentation and probability theory with undirected graphical models and in particular with the graphical model of Boltzmann machines. This combination of argumentation and graphical models allows us to relax assumptions on the probabilistic independences amongst arguments, and to obtain a closed-form representation of the distribution underlying observed labellings of arguments. By doing so, we can learn distributions of labellings from a set of cases (in an unsupervised manner) and sample labellings according to the learned distribution. We focus on generative learning and we thus propose a labelling random walk allowing us to generate grounded labellings with respect to the learned distribution. To make our ideas as widely applicable as possible, we investigate the use of Boltzmann machines (BMs) with the common abstract setting for argumentation of Dung (1995) where the content of arguments is left unspecified, hence we do not deal with structured argumentation in this paper, nor with sub-arguments (see Riveret et al., 2015 for a development of this paper which accounts for sub-arguments).

Since the model of BMs are commonly interpreted as a model of neural networks, we are moving towards the construction of neuro-argumentative systems based on a principled account of probabilistic argumentation. Moreover, as the graphical model of restricted Boltzmann machines (RBMs) can also be interpreted as a product of experts (PoE), our contribution can also be interpreted as a probabilistic model of an assembly of experts judging the status of arguments in an unanimous manner.

We investigate thus ‘argumentative Boltzmann machines’ meant to learn a function producing a probability distribution over status of arguments and to produce possible justified or rejected arguments sampled according to the learned distribution. As proposed by Mozina et al. (2007) and d’Avila Garcez et al. (2014), argumentation appears as a guide to learn hypotheses, that is, some background knowledge is assumed to be encoded into an argumentative structure that shall constrain the learning and the labelling of argumentative hypotheses. Structure learning is not addressed in this paper.

*Outline.* The paper is organised as follows. The motivations and possible applications of our proposal are developed in Section 2. The abstract argumentation is introduced in Section 3. We associate to it a probabilistic setting with a discussion with respect to probabilistic graphical models in Section 4, and we develop the probabilistic setting into an energy-based model in Section 5. We overview BMs in Section 6, and we propose ‘argumentative

Boltzmann machines' in Section 7. We give some experimental insights in Section 8 before concluding.

## 2. Motivations and applications

*The first motivation* of the present work originates from the observation that investigations into probabilistic argumentation had little consideration for the probabilistic dependences amongst arguments and for learning, cf. Riveret, Rotolo, and Sartor (2012b) and Sneyers et al. (2013).

Roth et al. (2007) used a probabilistic instantiation of an abstract argumentation framework in Defeasible Logic (Governatori, Maher, Billington, and Antoniou, 2004), with the focus on determining strategies to maximize the chances of winning legal disputes. In this work, the probability of the defeasible status of a conclusion is the sum of cases where this status is entailed. A case is a set of premises which are assumed independent. In the same line of research, Riveret et al. (2007) investigated a probabilistic variant of a rule-based argumentation framework proposed by Prakken and Sartor (1997). They looked at the probability to win a legal dispute, and thus on the probability that arguments and statements get accepted by an adjudicator. The proposal focused on the computation of probabilities of justification reflecting moves in a dialogue. They emphasized a distinction between the probability of the event of an argument and the probability of justification of an argument. The proposal assumes the independence amongst premises and arguments so that the probability of an argument is the product of the probability of its premises. Riveret et al. (2012a) and Riveret et al. (2012b) further developed the setting of Roth et al. (2007) to a rule-based argumentation framework by associating a potential to every sub-theory induced by a larger hypothetical theory. The potentials are then learned using reinforcement learning. Our proposal is inspired by this potential-based model of probabilistic argumentation, but for abstract argumentation combined with energy-based undirected graphical models, and in particular with the graphical model of BMs.

Sneyers et al. (2013) tackled a similar setting of Riveret et al. (2007) (but without the concern of reflecting the structure of dialogues) by defining a probabilistic argumentation logic and implementing it with the language CHRiSM (Sneyers, Meert, Vennekens, Kameya, and Sato, 2010), a rule-based probabilistic logic programming language based on constraint handling rules (CHR) (Fruhwirth, 2009) associated with a high-level probabilistic programming language called PRISM (Sato, 2008). This language must not be confused with the system PRISM for probabilistic model checking developed by Hinton et al. (2006). They discussed how it can be seen as a probabilistic generalization of the defeasible logic proposed by Nute (2001), and showed how it provides a method to determine the initial probabilities from a given body of precedents. In CHRiSM, rules have an attached probability and each rule is applied with respect to its probability: rules are probabilistically independent.

Dung and Thang (2010) extended Dung's abstract argumentation framework by investigating a probabilistic assumption-based argumentation (PABA) framework (Bondarenko et al., 1997) for jury-based dispute resolution. Every juror is associated with a probability space where a possible world can be interpreted as a set of arguments. The probability of an argument with respect to a juror and the grounded semantic is the sum of the probability of the juror's worlds entailing this argument. Thus, every juror determines the probability weights of the arguments. The probability space of every juror is further specified by mapping it to a PABA framework where some probabilistic parameters (appearing in some rules) define the possible worlds.

Li et al. (2011) proposed an extension of Dung's (1995) framework to build probabilistic argumentation frames by associating probabilities with the events of arguments and defeats. The

sample space is constituted by the possible argumentation sub-graphs induced from a probabilistic argumentation graph. Assuming the independence of arguments and defeats, the joint probability of the events of arguments and defeats induced from a probabilistic graph is then defined using the product rule for independent events. The probability of a set of arguments justified is defined as the marginal probability over the sub-graphs justifying this set of arguments. This setting is thus in tune with the setting used by Roth et al. (2007) where the abstract argumentation framework is instantiated in Defeasible Logic (Governatori et al., 2004) and when the sample space of sub-graphs is mapped to a sample space of cases, or with the proposal of Dung and Thang (2010) when the sample space of sub-graphs is mapped to the sample space as a set of arguments of a juror. Li et al. proposed to tackle the exponential complexity of the sample space by a Monte-Carlo algorithm to approximate the probability of a set of justified arguments, arguments being assumed independent. Li developed her work (see Li, 2015) to account for dependences amongst arguments when they share the same premises, however, these dependences remain logic and not probabilistic.

The abstract setting of Li et al. (2011) was followed by several investigations. For example, Fazzinga et al. (2013) observed that the Monte-Carlo techniques proposed by Li et al. (2011) can be improved, and they showed that computing the probability that a set of arguments is a set of justified arguments is either *P*TIME or  $FP^{\#P}$ -complete depending on the semantics. In a similar line of research, Dondio (2014) tackled the complexity of the sample space of the probabilistic framework proposed in (Li et al., 2011) by investigating a recursive algorithm to compute the probability of acceptance of each argument under grounded and preferred semantics. The underlying assumptions are that the probability of every argumentation sub-graph is known, or that the arguments are probabilistically independent, or that the premises are independent as in (Riveret et al., 2007).

Another view on probabilistic argumentation focuses on an interpretation of probabilities as subjective probabilities meant to represent the degree to which an argument is believed to hold. For example, Hunter (2013) developed the idea of a probability value attached to every argument, and investigated this idea for arguments that are constructed using classical logic. Then various kinds of inconsistency arising within the framework are studied. Thimm (2012) also focused on an interpretation of probabilities as subjective probabilities, and investigated probability functions that agree with some intuitions about the interrelationships of arguments and attacks. In these views, instead of differentiating the probability of the event of an argument and the probability of the justification of this argument, the probability of an argument is interpreted as its probability of its justification status. Hunter and Thimm (2014) further consolidated this interpretation according to which the status of arguments can be labelled in a similar way than conventional formal argumentation by assuming some properties on arguments probabilities. Baroni et al. (2014) observed that, though Hunter and Thimm focused on an interpretation of probabilities as subjective probabilities, Hunter and Thimm used Kolmogorov axioms instead of a probability theory more akin to a subjective probability setting. For this reason, Baroni et al. advocated and used the theory of De Finetti (1974) for an epistemic approach similar to Hunter (2013) and Thimm (2012).

In this paper, we favour a frequentist interpretation of probabilities, instead of an interpretation of probabilities as subjective probabilities. The subjective approach is interesting when the domain is small and when objectivity is not important, but a frequentist approach where probabilities are statically computed or learned is certainly more appropriate for larger domains or more objective measures. However, since frequency ratios may also be interpreted in subjective probabilities and vice versa, we do not discard an interpretation of probabilities as subjective probabilities for our framework, in particular when the framework is understood from the perspective of the model of products of experts.

Whatever the interpretation of probabilities, we note that the extraction of probability measures is rather neglected in the above proposals and there is no consideration for learning. An exception is the work by Sneyers et al. (2013) which proposes a solution to determine the initial probabilities from a given body of precedents (but rules are probabilistically independent). Riveret et al. (2012b) also investigated an approach using reinforcement learning to learn the optimal probability of using some arguments by an agent.

Furthermore, we also note that all the above works assume that either the probability of every argumentation sub-graph is known (for theoretical purposes) or that the arguments are probabilistically independent (for practical purposes), but such assumptions may not hold in the general case and for many applications. For these reasons, we aim at developing a computational probabilistic argumentation framework able to handle probabilistic dependences amongst arguments and learn probability measures from examples.

*The second motivation* of our proposal stems from the observation that logics, principled probabilistic and statistical approaches along machine learning have been largely studied to learn from uncertain knowledge and to perform inferences with this knowledge. In particular, Statistical Relational Learning (SRL) (Getoor and Taskar, 2007) is the field at this crossroad, with successful integrations, see, for example, probabilistic relational models (Getoor et al., 2007) and Markov logic networks (Richardson and Domingos, 2006). In these approaches, logics are used to structure in a qualitative manner the probabilistic relations amongst entities. Typically, (a subset of) first-order logic formally represents a qualitative knowledge describing the structure of a domain in a general manner (using universal quantification) while techniques from graphical models (Koller and Friedman, 2009) such as Bayesian networks (Getoor et al., 2007) or Markov networks (Richardson and Domingos, 2006) are applied to handle probability measures on the structured entities. Though SRL approaches focus on capturing data in its relational form and we are only dealing with an abstract and flat data representations (induced by our setting of abstract argumentation), we are inspired by these approaches consisting in combinations of graphical models and logic-based systems, and we propose that investigations for probabilistic argumentation shall benefit from the use of graphical models too.

In this paper, we will focus on abstract argumentation, and thus we need to select a probabilistic graphical model appropriate for our purposes. In SRL systems, the probabilistic dependences are usually represented by the graphical structure of the probabilistic graphical model associated to the logic-based knowledge. These relations of (in)dependence amongst entities of the system are specified or induced by the logic-based knowledge, or given by domain experts or learned from examples. In the case of probabilistic argumentation, though arguments are usually assumed to have no other dependence relations other than the relations of attacks and eventual supports, we will argue that the argumentative structure and the probabilistic structure have to be separated (see Section 4). We will assume that the dependences are not specified (from a human operator for instance): so we will focus on a particular class of graphical models called Boltzmann machines where any dependence amongst arguments is possible. These machines are a Monte Carlo version of Hopfield networks inspired by models in statistical physics, and they can be interpreted as a special case of products of experts or Markov networks (see Section 6). The model of BMs appears thus as an appealing theoretical model which has shown to be useful for practical problems and which paves the way to some multi-modal machines (combining audio, video and symbolic modes for example), but this model has not been considered so far to tackle probabilistic and learning settings of formal argumentation.

Learning is commonly approached in a discriminative or a generative way. The difference holds in that a generative approach deals with the joint probability over all variables in a system and handles it to perform other tasks like classification, while discriminative approaches are meant to directly capture the conditional probability for classification. More precisely, if a system



is featured by  $N$  variables  $(x_1, \dots, x_N)$ , the generative approach will account for the full joint distribution  $p(x_1, \dots, x_N)$ , while the discriminative approach will attempt to capture a conditional probability like  $p(x_N | x_1, \dots, x_{N-1})$  where  $x_N$  is a classifying variable. A generative model can be handled to deal with tasks performed by discriminative approaches, or it can be used in a straightforward manner to simulate (i.e. generate, or sample) configurations of a system. In this paper, we will focus on the generative approach. So, given examples of the form  $(x_1, \dots, x_N)$ , we will train an argumentative BM to account for the full joint distribution  $p(x_1, \dots, x_N)$  of the examples in a generative manner, paving the way to the manipulation of the learned joint distribution to account for a conditional distribution. Since we are interested by generative learning, we will focus on a labelling random walk which allows us to generate grounded labellings according to a learned probability distribution.

Whilst training datasets in state-of-the-art SRL systems are normally relational, a training example for BMs is typically a binary vector. For our purposes and since we are dealing with abstract argumentation, a training case will be a labelling of a set of arguments, that is, every argument will be associated a label to indicate whether its status is justified, rejected, undecided or unexpressed. We will show that the space complexity of the sample space for probabilistic argumentation can be dramatically reduced when using BMs, and a compact model can be learned at run time while producing labellings.

*The third motivation* of our work regards the construction of neuro-symbolic agents. Neuro-symbolic agents join the strength of neural networks and logics, see, for example, d'Avila Garcez et al. (2008). Neural networks offer robust learning with the possibility of massive parallel computation and have successfully shown to grasp information and uncertainty where symbolic approaches failed. However, as networks suffer from their 'black box' nature, logic formalisms shall bring intelligible knowledge representation and reasoning into the networks with explanatory ability, thereby facilitating the transmission of learned knowledge to other agents.

As artificial neural networks are inspired by natural nervous systems, we propose to associate to them argumentation as a natural account of human defeasible reasoning. 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. We will propose to build neuro-symbolic agents based on BMs and formal argumentation. As a different Monte Carlo sampling approach to BMs are possible, we will propose a sampling of labellings allowing the use of variants of training techniques.

Interestingly, as BMs feature a random walk, and because argumentation is an intuitive account of common sense reasoning, our proposal may be interpreted as an abstract account of common sense reasoning appearing in a random walk of thoughts, constrained by some logical principles of argumentation.

The combination of argumentation frameworks and neural networks is not new. For example, d'Avila Garcez (2005) presented a combination of value-based argumentation frameworks and neuro-symbolic learning systems by translating argumentation frameworks into neural networks. In the same line of research, d'Avila Garcez et al. (2014) generalised the model in d'Avila Garcez et al. (2005) to translate standard and non-standard argumentation frameworks into standard connectionist networks, with application to legal inference and decision-making. We share a connectionist approach for argumentation, but they explored a combination of directed neural network and argumentation with no consideration for any probabilistic setting for argumentation, while we opt for an undirected network (BM) with a principled probabilistic setting.

*As to the applications*, argumentation along probability theory and machine learning is naturally well adapted to account for argumentative domains like in Law, but they can also be used in

many domains where phenomena can be structured in a dialectic way or where some defeasible reasoning can be advantageous to reason about these phenomena. This includes domains where general rules can be set up to account for the default cases, and rules of exceptions are defined to undermine default arguments.

It is fundamental to remark that labellings of arguments always result from an argumentative activity. Furthermore, since argumentation technologies are still a matter of research before its eventual adoption in operational socio-technical systems, datasets of labelled arguments are clearly uncommon nowadays. Thus, we anticipate systems where data as traditionally presented to learning systems can be seen as the premises of arguments so that arguments can be built from these data. And since argumentation is meant to ease communication along a form of collective defeasible reasoning in multi-agent settings, agents shall communicate arguments and eventually their labellings. This view underlies the construction of the Semantic Web with argumentation technologies as investigated for example by Torroni et al. (2007, 2009), Schneider et al. (2013) and Rahwan (2008).

The model of BMs offers a compact representation of a distribution of training datasets, and thus such a graphical model is particularly interesting when the set of possible states of a system is too large to be fully considered. This is typically the case of multi-modal learning from semi-structured data mixing structured symbolic elements with unstructured content like texts or images and others. For example, an argumentation may be associated with contextual information or rhetoric features, and a machine shall learn patterns or correlations amongst all these pieces of information, including the status of arguments. Even if a set of all possible cases is available, one may prefer a compact representation instead of handling a large dataset, as a data repository is often time consuming and error prone to manage, or just rebutting in practice when one desires to quickly build simple or robust applications. The dataset of argumentations may also be too large to be embarked into some autonomous mobile agents such as robots, drones or embedded systems with bounded resources. In this case, a closed-form representation of the distribution underlying the dataset may be necessary. Eventually, if a probabilistic argumentative model has to be communicated and exploited, then it may be preferable to communicate such closed-form representation rather than the whole dataset.

In multi-agent simulations, stochastic behaviours of heterogeneous agents may be simulated by sampling behaviours from argumentative machines, where argumentation shall provide scrutiny on the simulated behaviours. As some non-monotonic logics instantiating Dung's abstract argumentation framework are proposed to model normative multi-agent systems (see e.g. Sartor, 2005; Governatori et al., 2005) with some implementations Riveret et al., 2012a,b), the model of BMs combined with abstract argumentation paves the way to simulations combining neural agents and institutions. In 'living' simulators for which data shall be streamed into (see e.g. Paolucci et al., 2013), a compact model shall be learned 'on the fly' while producing possible labellings sampled according to the learned distribution. More generally, learning a compact model on the fly while sampling labellings can also be interesting for so-called complex event processing systems capturing data streams.

### 3. Abstract argumentation framework

In this section, we introduce our abstract argumentation framework. We begin with the definition of an argumentation graph following Dung (1995).

**DEFINITION 3.1 (Argumentation graph)** An argumentation graph is a pair  $\langle \mathcal{A}, \rightsquigarrow \rangle$  where  $\mathcal{A}$  is a set of arguments, and  $\rightsquigarrow \subseteq \mathcal{A} \times \mathcal{A}$  is a binary relation of attack.



Figure 1. Pictorial representation of an argumentation graph. Argument B attacks argument C, arguments C and D attack each other.

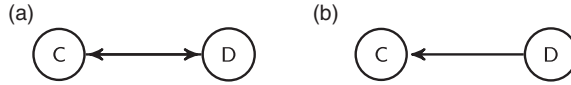


Figure 2. The graph (a) is a sub-graph of the graph in Figure 1, while the graph (b) is not one of its sub-graphs.

*Example 3.2* The argumentation graph  $\langle\{B, C, D\}, \{B \rightsquigarrow C, C \rightsquigarrow D, D \rightsquigarrow C\}\rangle$  is illustrated in Figure 1.

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

We say that a sub-graph  $\mathcal{H}$  of an argumentation graph  $\mathcal{G}$  is induced if, for any pair of arguments  $A$  and  $B$  in  $\mathcal{H}$ , the attack  $A \rightsquigarrow B$  is an attack of  $\mathcal{H}$  if and only if this attack is an attack of  $\mathcal{G}$ . Thus,  $\mathcal{H}$  is an induced sub-graph of  $\mathcal{G}$  if it has exactly the attacks that appear in  $\mathcal{G}$  over the same set of arguments (see Figure 2).

Once we have an argumentation graph, we can compute the set of arguments that are justified or rejected, that is, those arguments that shall survive or not to possible attacks. To do that, we need semantics. Many semantics exist and, for our purposes, we will consider Dung’s grounded semantics (Dung, 1995) by labelling arguments as in Baroni et al. (2011), but slightly adapted to our purposes. Accordingly, we will distinguish  $\{\text{in}, \text{out}, \text{un}\}$ -labellings and  $\{\text{in}, \text{out}, \text{un}, \text{off}\}$ -labellings. In a  $\{\text{in}, \text{out}, \text{un}\}$ -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 discarded. The label ‘un’ marks the status of the argument as undecided. We introduce two novel labels: ‘on’ and ‘off’ to indicate whether an argument is expressed or not. We say that an argument is expressed if it is labelled on, or either in, out or un, otherwise it is not expressed, that is, it is labelled off. So, in a  $\{\text{on}, \text{off}\}$ -labelling or in a  $\{\text{in}, \text{out}, \text{un}, \text{off}\}$ -labelling, the label ‘off’ indicates that the argument is not expressed.

**DEFINITION 3.3 (Labellings)** Let  $\mathcal{G}$  be an argumentation graph.

- A  $\{\text{on}, \text{off}\}$ -labelling of  $\mathcal{G}$  is a total function  $l : \mathcal{A}_{\mathcal{G}} \rightarrow \{\text{on}, \text{off}\}$ .
- A  $\{\text{in}, \text{out}, \text{un}\}$ -labelling of  $\mathcal{G}$  is a total function  $l : \mathcal{A}_{\mathcal{G}} \rightarrow \{\text{in}, \text{out}, \text{un}\}$ .
- A  $\{\text{in}, \text{out}, \text{un}, \text{off}\}$ -labelling of  $\mathcal{G}$  is a total function  $l : \mathcal{A}_{\mathcal{G}} \rightarrow \{\text{in}, \text{out}, \text{un}, \text{off}\}$ .

Notice that the labelling functions are denoted with a lower case, reserving the upper case to denote random labellings in the probabilistic setting (as we will see in Section 4). We write  $\text{in}(l)$  for  $\{A \mid l(A) = \text{in}\}$ ,  $\text{out}(l)$  for  $\{A \mid l(A) = \text{out}\}$ ,  $\text{un}(l)$  for  $\{A \mid l(A) = \text{un}\}$ ,  $\text{on}(l)$  for  $\{A \mid l(A) = \text{on}\}$  and  $\text{off}(l)$  for  $\{A \mid l(A) = \text{off}\}$ . In the remainder, a  $\{\text{in}, \text{out}, \text{un}\}$ -labelling  $l$  will be represented as a tuple  $(\text{in}(l), \text{out}(l), \text{un}(l))$ , and a  $\{\text{in}, \text{out}, \text{un}, \text{off}\}$ -labelling  $l$  as a tuple  $(\text{in}(l), \text{out}(l), \text{un}(l), \text{off}(l))$ . Next *complete labellings* are defined by constraints amongst the labels in and out.

**DEFINITION 3.4 (Complete  $\{\text{in}, \text{out}, \text{un}\}$ -labelling)** Let  $\mathcal{G}$  be an argumentation graph, a complete  $\{\text{in}, \text{out}, \text{un}\}$ -labelling of  $\mathcal{G}$  is a  $\{\text{in}, \text{out}, \text{un}\}$ -labelling such that for every argument  $A$  in  $\mathcal{A}_{\mathcal{G}}$  it holds that:



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

An argumentation graph may have several complete  $\{\text{in}, \text{out}, \text{un}\}$ -labellings. Given a graph, we will focus in the remainder on the unique complete labelling with the smallest set of labels **in** (or equivalently the largest set of labels **un**) (Dung, 1995; Baroni et al., 2011).

**DEFINITION 3.5** (Grounded  $\{\text{in}, \text{out}, \text{un}\}$ -labelling) Let  $\mathcal{G}$  be an argumentation graph and  $l$  a complete  $\{\text{in}, \text{out}, \text{un}\}$ -labelling of  $\mathcal{G}$ . A labelling  $l$  is a grounded  $\{\text{in}, \text{out}, \text{un}\}$ -labelling of  $\mathcal{G}$  if  $\text{in}(l)$  is minimal (w.r.t. set inclusion) amongst all complete  $\{\text{in}, \text{out}, \text{un}\}$ -labellings of  $\mathcal{G}$ .

An algorithm for generating the grounded  $\{\text{in}, \text{out}, \text{un}\}$ -labelling of an argumentation graph is given in Algorithm 1 (Modgil and Caminada, 2009). 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 which remained unlabelled (line 7).

---

**Algorithm 1** Computation of the grounded  $\{\text{in}, \text{out}, \text{un}\}$ -labelling of an argumentation graph.

---

```

1: input Argumentation graph  $\mathcal{G}$ 
2:  $l_0 = (\emptyset, \emptyset, \emptyset)$ 
3: repeat
4:    $\text{in}(l_{i+1}) \leftarrow \text{in}(l_i) \cup \{A \mid A \text{ is not labelled in } l_i, \text{ and } \forall B : \text{if } B \text{ attacks } A \text{ then } B \in \text{out}(l_i)\}$ 
5:    $\text{out}(l_{i+1}) \leftarrow \text{out}(l_i) \cup \{A \mid A \text{ is not labelled in } l_i, \text{ and } \exists B : B \text{ attacks } A \text{ and } B \in \text{in}(l_{i+1})\}$ 
6: until  $l_i = l_{i+1}$ 
7: return  $(\text{in}(l_i), \text{out}(l_i), A - \text{in}(l_i) \cup \text{out}(l_i))$ 

```

---

This algorithm is sound because the resulting  $\{\text{in}, \text{out}, \text{un}\}$ -labelling  $l$  is complete and  $\text{in}(l)$  is minimal, thus the labelling  $l$  is a grounded  $\{\text{in}, \text{out}, \text{un}\}$ -labelling. At each iteration, one argument at least is labelled, otherwise the algorithm terminates. Therefore, when the input is a graph of  $N$  arguments, then there is necessarily less than  $N$  iterations. For each iteration, the status of  $N$  arguments has to be checked with respect to the status of  $N$  arguments in the worst case. Consequently, the time complexity of Algorithm 1 is polynomial in the worst case.

We introduced  $\{\text{in}, \text{out}, \text{un}, \text{off}\}$ -labellings in Definition 3.3 by extending the common  $\{\text{in}, \text{out}, \text{un}\}$ -labelling with unexpressed arguments, we present now *grounded*  $\{\text{in}, \text{out}, \text{un}, \text{off}\}$ -labellings. The substantial underlying idea is that only expressed arguments can be refuted, and only expressed arguments can effectively attack other arguments.

**DEFINITION 3.6** (Grounded  $\{\text{in}, \text{out}, \text{un}, \text{off}\}$ -labelling) Let  $\mathcal{G}$  be an argumentation graph and  $\mathcal{H}$  an induced sub-graph of  $\mathcal{G}$ . A grounded  $\{\text{in}, \text{out}, \text{un}, \text{off}\}$ -labelling of  $\mathcal{G}$  is a  $\{\text{in}, \text{out}, \text{un}, \text{off}\}$ -labelling such that:

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

An argumentation graph has a unique grounded  $\{\text{in}, \text{out}, \text{un}\}$ -labelling, but it has as many grounded  $\{\text{in}, \text{out}, \text{un}, \text{off}\}$ -labellings as sub-graphs. See Figure 3 for examples of grounded  $\{\text{in}, \text{out}, \text{un}, \text{off}\}$ -labellings.

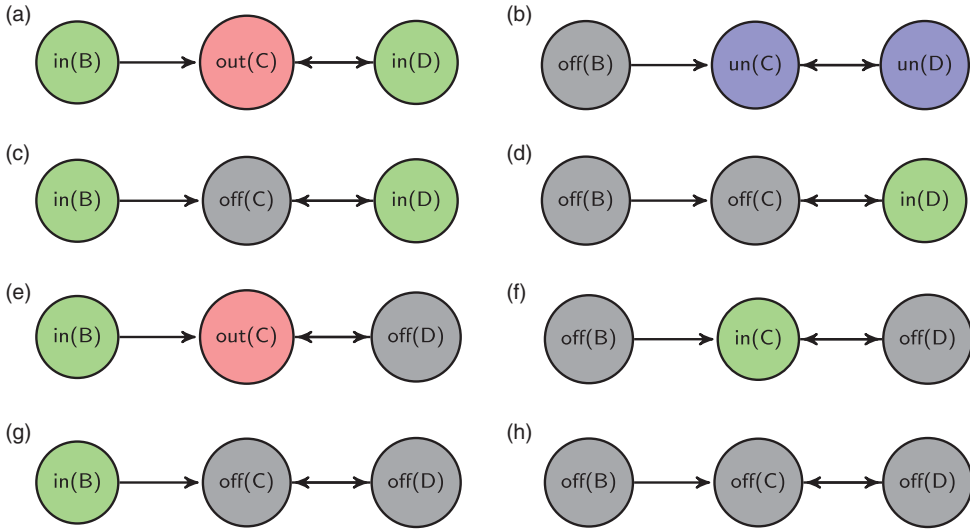


Figure 3. Grounded {in, out, un, off}-labellings with a probability possibly different to 0, when considering the grounded {in, out, un, off}-labelling for the probabilistic argumentation frame as drawn in Figure 1.

As for notational matters, the sets of labellings of an argumentation graph  $\mathcal{G}$  will be denoted  $\mathcal{L}_{\mathcal{G}}$ . A complete {in, out, un, off}-labelling will be abbreviated as {in, out, un, off}<sup>c</sup>-labelling, and a grounded {in, out, un, off}-labelling as {in, out, un, off}<sup>g</sup>-labelling. By doing so, we can denote the set of  $X$ -labellings of a graph  $\mathcal{G}$  as  $\mathcal{L}_{\mathcal{G}}^X$ , and each set will basically constitute a sample space of our probabilistic setting for argumentation. Considering a  $X$ -labelling, if a labelling of arguments is not a  $X$ -labelling, then it is an *illegal* labelling, else it is a *legal* labelling.

#### 4. Probabilistic abstract argumentation framework

In this section, we introduce our probabilistic setting for the abstract argumentation framework as given in the previous section. Notice that there are many ways to define the probability space of a setting for probabilistic argumentation and the choice of a probability space shall be made in function of the envisaged applications. Our choice was conducted by the promotion of simplicity and to match our use of the graphical model of BMs for probabilistic argumentation towards the construction of neuro-argumentative systems.

So for our purposes, we define a sample space as the set of labellings of a *hypothetical argumentation frame* (or simply called a hypothetical frame) which is an argumentation graph. Now, when building a probabilistic setting, we have to make the distinction between between an impossible event (i.e. the event does not belong to the sample space), and an event with probability 0. Similarly, given a hypothetical argumentation frame  $\mathcal{G}$  and a  $X$ -labelling (e.g. a {on, off}-labelling or a grounded {in, out, un, off}-labelling or any other {in, out, un, off}-labelling), there is the choice between either (i) defining the sample space as the set of the  $X$ -labellings of  $\mathcal{G}$ , or (ii) defining the sample space as the set of any labelling of  $\mathcal{G}$  and the probability will be 0 for any labelling which is not a  $X$ -labelling of  $\mathcal{G}$ . In fact, this distinction does not really matter in practice, but it turns out that the second definition of the sample space shall simplify the use of factors (as we will do soon), and it is the reason we will favour it here.

**DEFINITION 4.1 (Probabilistic argumentation frame)** A probabilistic argumentation frame is a tuple  $(\mathcal{G}, X, (\Omega, F, P))$  such that  $\mathcal{G}$  is an argumentation graph (called the hypothetical

argumentation frame),  $X$  indicates the type of the considered  $X$ -labelling and  $(\Omega, F, P)$  is a probability space such that:

- the sample space  $\Omega$  is the set of labellings with respect to  $\mathcal{G}$ , i.e.  $\Omega = \mathcal{L}_{\mathcal{G}}$ ,
- the  $\sigma$ -algebra  $F$  is the power set of  $\Omega$ ,
- the function  $P$  from  $F(\Omega)$  to  $[0, 1]$  is a probability distribution (satisfying Kolmorov axioms), such that for any labelling  $l$  not in the set of  $X$ -labellings of  $\mathcal{G}$ , the probability of  $l$  is 0, i.e.  $\forall l \in \mathcal{L}_{\mathcal{G}} \setminus \mathcal{L}_{\mathcal{G}}^X, P(\{l\}) = 0$ .

In the remainder, we may write  $P(\{l\})$  as  $P(l)$  for the sake of notational clarity. The proposed probabilistic setting is generic in the sense that it can host other types of labellings such as complete {in, out, un, off}-labellings, preferred {in, out, un, off}-labellings (not presented here), etc. In the remainder, we will focus on the sample spaces of the {on, off}-labellings and grounded {in, out, un, off}-labellings.

*Example 4.2 (continued)* Assume a hypothetical argumentation frame as drawn in Figure 1. Considering the grounded {in, out, un, off}-labelling for our probabilistic argumentation frame, the grounded {in, out, un, off}-labellings with a probability possibly different to 0 are shown in Figure 3. Considering the {on, off}-labelling for our probabilistic argumentation frame, the {on, off}-labellings of the corresponding sample space are drawn in Figure 4.

Remark that a probabilistic argumentation frame where the labelling is a {on, off}-labelling boils down to the case where the sample space is the set of sub-graphs of the hypothetical argumentation frame, cf. Li et al. (2011), Fazzinga et al. (2013), and Dondio (2014). Moreover, since any legal {on, off}-labelling can be trivially mapped to one grounded {in, out, un, off}-labelling and only one (as we can visualise in Figure 4), a probabilistic argumentation frame with {on, off}-labellings is equivalent to one with grounded {in, out, un, off}-labellings because they shall give the same probability measures for equivalent events.

As we have defined our probabilistic space, we can consider random variables, that is, functions (traditionally defined by an upper case letter such as  $X$ ) from the sample space  $\Omega$  into

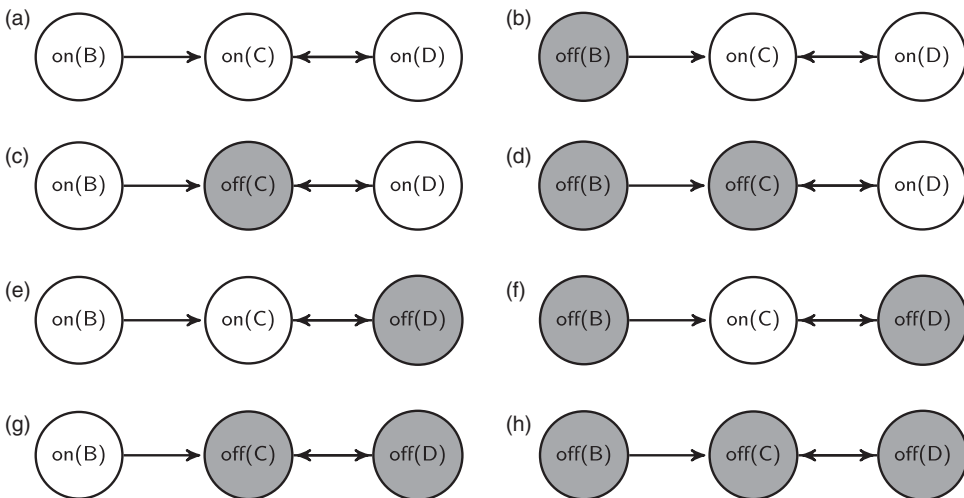


Figure 4. Sample space as the set of {on, off}-labellings.

another set of elements. So, we will associate to every argument  $A$  some random variables, called *random labellings*, to capture the random labelling of the argument.

Random labellings can take different forms. An option consists in considering *binary random labellings*, that we denote  $L_{A,\text{on}}, L_{A,\text{off}}, L_{A,\text{in}}, L_{A,\text{out}}, L_{A,\text{un}}$  such that  $L_{A,\text{lab}} : \Omega \rightarrow \{0, 1\}$  with  $\text{lab} \in \{\text{on}, \text{in}, \text{out}, \text{un}, \text{off}\}$ . So, if the set of considered labels is  $\{\text{on}, \text{off}\}$  for instance, then we write  $L_{A,\text{on}} = 1$  as a shorthand for the event  $\{\{l\} \in \Omega \mid L_{A,\text{on}}(l) = 1\}$  according to which the argument  $A$  is labelled **on**. An alternative to binary random labellings consists in categorical random variables, called a *categorical random labellings* and denoted  $L_A$ , from  $\Omega$  into a set of labels. So, we write  $L_A = \text{on}$  as a shorthand for the event  $\{\{l\} \in \Omega \mid L_A(l) = \text{on}\}$ , and since an argument can be labelled **on** or **off** in a  $\{\text{on}, \text{off}\}$ -labelling, we can readily note that

$$P(L_A = \text{on}) + P(L_A = \text{off}) = 1. \tag{1}$$

The use of binary random labellings or categorical random labellings is rather a notational matter, but since categorical random labellings offer a more compact notation, we will favour them in the remainder of this section. We will denote  $\text{Val}(L_A)$  the set of values that a categorical random labelling can take, for example,  $\text{Val}(L_A) = \{\text{on}, \text{off}\}$  in the case of  $\{\text{on}, \text{off}\}$ -labellings. When the context does not rise any ambiguity, we shall abbreviate the categorical random labelling  $L_A$  by simply  $A$  (in italics).

We use boldface type to denote sets of random labellings, thus  $\mathbf{L}$  denotes a set of random labellings  $\{L_{A_1}, \dots, L_{A_N}\}$ . The joint distribution over a set  $\mathbf{L} = \{L_{A_1}, L_{A_2}, \dots, L_{A_N}\}$  of random labellings is formally denoted  $P(\{L_{A_1}, L_{A_2}, \dots, L_{A_N}\})$ , but we will write it  $P(L_{A_1}, L_{A_2}, \dots, L_{A_N})$ . We also use a lower boldface type to denote assignment to a (set of) random variable(s), in particular  $\mathbf{l}$  will denote an assignment of a set  $\mathbf{L}$  of random labellings.

Given a set  $\mathbf{L}$  of random labellings, we consider the standard definition of a *factor* as a function, denoted  $\phi$ , from the possible set of assignments  $\text{Val}(\mathbf{L})$  to strictly positive real numbers  $\mathbb{R}^+$ . The set  $\mathbf{L}$  is called the scope of the factor  $\phi$ . On this basis, we write the joint distribution of random labellings  $L_{A_1}, \dots, L_{A_N}$  as a distribution parametrised by a set of factors  $\Phi = \{\phi_1(\mathbf{L}_1), \dots, \phi_K(\mathbf{L}_K)\}$ :

$$P_\Phi(L_{A_1}, L_{A_2}, \dots, L_{A_N}) = \frac{\prod_i \phi_i(\mathbf{L}_i)}{\mathcal{Z}_\Phi}, \tag{2}$$

where  $\mathcal{Z}_\Phi$  is normalising the distribution:

$$\mathcal{Z}_\Phi = \sum_{L_{A_1}, \dots, L_{A_N}} \prod_i \phi_i(\mathbf{L}_i). \tag{3}$$

A factor can be thought as an expert putting some credits on assignments, we will return on this point when introducing BMs in Section 6.

If the scope of a factor contains two random labellings, then there is a direct probabilistic influence between these two random labellings. In this view, it is common to visualise dependences by relating a factorisation with a graphical structure, typically such as a Markov random field (MRF), also known as Markov network, where nodes represent random variables and arcs indicate dependences amongst these variables: in this case, a distribution  $P_\Phi$  with  $\Phi = \{\phi_1(\mathbf{L}_1), \dots, \phi_K(\mathbf{L}_K)\}$  factorises over a Markov network  $H$  if each  $\mathbf{L}_i$  is a complete sub-graph of  $H$ .

It is also common to visualise the factorisation under the graphical model of factor graphs, which is an alternative to Markov networks to explicit the factorisation. In this perspective, a factor graph for a set of random labellings  $\mathbf{L} = (L_1, L_2, \dots, L_N)$  and a set of factor functions  $\phi_1, \dots, \phi_K$  is a bipartite graph on a set of nodes corresponding to the variables, and a set

$L_B$	$L_C$	$L_D$	$\phi_1(L_B, L_C, L_D)$	$L_D$	$L_E$	$L_F$	$\phi_2(L_D, L_E, L_F)$
on	on	on	1	on	on	on	500
on	on	off	100	on	on	off	100
on	off	on	10	on	off	on	10
on	off	off	10	on	off	off	10
off	on	on	50	off	on	on	50
off	on	off	1	off	on	off	1
off	off	on	1	off	off	on	1
off	off	off	1	off	off	off	1

Figure 5. Tables corresponding to the factors  $\phi_1(L_B, L_C, L_D)$  and  $\phi_2(L_D, L_E, L_F)$ .

of nodes corresponding to the functions. Each function depends on a subset of the variables, and the corresponding function node is connected to the nodes corresponding to the subset of variables, that is, if  $\phi_k$  depends on  $L_n$ , there is an edge connecting the nodes representing  $\phi_k$  and  $L_n$ .

*Example 4.3* Consider an argumentation graph  $G$  with five arguments, say  $\mathcal{A}_G = \{B, C, D, E, F\}$  (the attack relations does not matter for our purposes), we can (arbitrarily) decompose the joint distribution of random labelling in two factors as follows:

$$P(L_B, L_C, L_D, L_E, L_F) = \frac{\phi_1(L_B, L_C, L_D) \cdot \phi_2(L_D, L_E, L_F)}{\mathcal{Z}_\phi}. \quad (4)$$

Suppose that the factors take their values as given in the tables of Figure 5.

From the given factors, we can compute the joint probability of any {on, off}-labelling of the argumentation graph. For example, consider the probability of the {on, off}-labelling where all the arguments are labelled on. Since  $\phi_1(L_B = \text{on}, L_C = \text{on}, L_D = \text{on}) = 0$  and  $\phi_2(L_D = \text{on}, L_E = \text{on}, L_F = \text{on}) = 50$ , we can compute the following joint probability (amongst others):

$$P(L_B = \text{on}, L_C = \text{on}, L_D = \text{on}, L_E = \text{on}, L_F = \text{on}) \approx 0. \quad (5)$$

Graphical models provide a compact and graphical representation of dependences to deal with joint probability distributions. For example in MRFs, two sets of nodes  $A$  and  $B$  are conditionally independent given a third set  $C$ , if all paths between the nodes in  $A$  and  $B$  are separated by a node in  $C$ . In graphical models, arcs are thus meant to represent assumptions on conditional independence relationships amongst variables, and on the basis of these assumptions joint probability can break down into manageable products of conditional probabilities.

In abstract argumentation graphs, nodes represent arguments and arrows represent attacks. Though an attack necessarily indicates some sort of influence on the probability of justification of an argument, the structure of an argumentation graph is not meant to account for conditional independence relationships amongst arguments in the probabilistic sense.

For example, we could suppose that argument **B** in the graph of Figure 6 is marginally independent of argument **D**. However, such assumption may not be acceptable in many cases. For instance, suppose that

- **C** supports the conclusion that visitors have no discounts;
- **B** supports the conclusion that a visitor has a discount because she or he is a student;
- **D** supports the conclusion that a visitor has a discount because she or he is under 25.

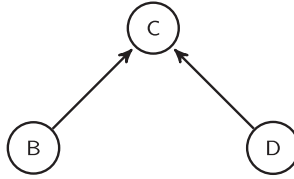


Figure 6. In this argumentation graph, arguments **B** and **D** are not necessarily independent.

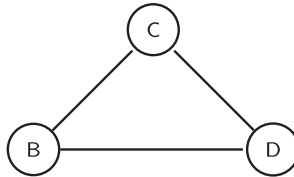


Figure 7. A MRF specifying explicit dependence amongst arguments **B**, **C** and **D**.

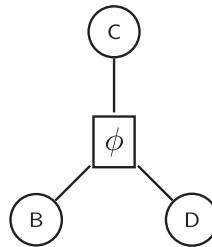


Figure 8. A (trivial) factor graph specifying the joint probability  $P(L_B, L_C, L_D)$ .

Clearly, the labelling of **B** and **D** should not be independent because many students are often under 25. Hence, this argumentation graph is not meant to account for dependence arcs as commonly understood in graphical models, and to capture the dependences, one may draw a Markov network as given in Figure 7, or a factor graph as in Figure 8. This example may be considered in a modified setting for abstract argumentation with some support relations (see e.g. Cayrol and Lagasquie-Schiex, 2013), so that arguments **B** and **C** are related by some support relations, specifying so some probabilistic dependences. In this view, we might induce a Markov network from an argumentation graph to capture the probabilistic dependences among the labels of arguments, but this matter is not addressed in this paper.

In this paper, we consider that graphs of probabilistic graphical models and argumentation graphs (and logical structures in general) are not meant to encode the same type of information (cf. Richardson and Domingos, 2006): graphs of probabilistic graphical models encode probabilistic dependence relations, while argumentation graphs encode logic constraints on the possible labellings amongst arguments. In this view, though graphs of probabilistic graphical models and argumentation carry distinct information, we see them as complementary, and an argumentation graph shall be associated with a probabilistic graphical model to specify probabilistic dependences.

If a Markov network is used as the probabilistic graphical model, then the dependencies can be given by a human operator, but they may also be discovered using some structure learning algorithms. An alternative is to introduce some hidden variables, represented by some binary hidden nodes in the Markov network, to account for unknown dependences. So assuming hidden



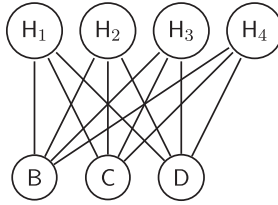


Figure 9. Graphical representation of pairwise graphical model with four hidden nodes of four hidden variables and three nodes of three categorical random labellings, such that there is no direct dependences between hidden variables or between random labellings.

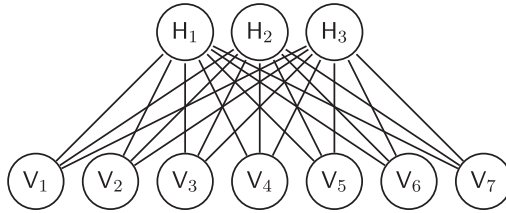


Figure 10. Graphical representation of a restricted BM with three hidden units and seven visible units. When the machine is interpreted as a neural network, each unit represents a neuron. When the machine is interpreted as a graphical model of a PoE, then each hidden unit represents an expert.

variables, we may consider an assignment of binary random labellings  $\mathbf{l}$  and the assignment of  $K$  binary hidden variables  $\mathbf{h} \in \{0, 1\}^K$  such that the probability of the overall assignment will have the following form:

$$P_{\Phi}(\mathbf{l}, \mathbf{h}) = \frac{1}{Z_{\Phi}} \prod_{k=1}^K \phi_k(\mathbf{l}, \mathbf{h}). \quad (6)$$

In particular, we will focus on a pairwise graphical model where there are no connections between nodes representing arguments labels, and between hidden nodes (Figures 9 and 10).

In such a bipartite graphical model, we may introduce a factor  $\phi_{ij}$  for every vertex between the node of a random labelling  $L_i$  and the node of a hidden variable  $H_j$ , so that Equation (6) can be rewritten as follows:

$$P_{\Phi}(\mathbf{l}, \mathbf{h}) = \frac{1}{Z_{\Phi}} \prod_{(l_i, h_j)} \phi_{ij}(l_i, h_j). \quad (7)$$

As we will see, such a graphical model corresponds in fact to a particular graphical model of BMs called *restricted Boltzmann machines* (RBM). Using this model, we shall benefit for efficient learning algorithms to capture a compact model of a set of training examples, which can be accommodated to our probabilistic argumentation setting to discard illegal labelling (with respect to grounded {in, out, un, off}-labellings). Interestingly, we will see that the model of restricted machines can be understood as a model of PoEs for our probabilistic argumentation setting, where each hidden node is possibly interpreted as an expert.

As the model of BMs is an energy-based model, we take the opportunity to present in the next section a counterpart energy-based model for our argumentation setting and its possible justification, and this will allow us to account for the settings investigated in Li et al. (2011), Fazzinga et al. (2013), and Dondio (2014) where arguments are probabilistically independent, as well as the setting of Riveret et al. (2012a,b) where the dependences amongst arguments are

captured by a potential (which is a loose terminology for the energy function) meant to be learned by reinforcement learning.

## 5. Energy-based argumentation

Energy-based models in probabilistic settings associate an energy to any configuration defined by the variables of a system, and then associate a probability to each configuration on the basis of energies (see e.g. LeCun et al., 2006 for a tutorial).

Making an inference with an energy-based model consists in comparing the energies associated with various configurations of the variables, and choosing the one with the smallest energy. The energies of remarkable configurations can be given by a human operator, or they can be learned using machine learning techniques. In particular, any illegal configuration shall be set with an infinite energy, so that such configurations will occur with probability zero.

Energy-based models date back from the early work of statistical physics, notably the work of Boltzmann and Gibbs, and have since inspired more abstract probabilistic setting in particular so-called log-linear models in probabilistic graphical models. So energy-based models appear not only as a probabilistic setting with a strong theoretical basis, but have been shown to be useful in many practical applications. We are thus inspired by works using energy-based models, and we investigate here such energy-based models for argumentation. Sometimes, energy-based models are also called potential-based models, and we may favour the latter denomination to the former when we deal with argumentation, because it appears to us more natural to talk about the potential of arguments than the energy of arguments. However, we prefer here to align ourselves with the standard terminology of energy-based models. In the setting of argumentation, the energy can be interpreted as an opposite measure of the credit we put into the argumentative status of arguments, for example, the compatibility amongst the status taken by the arguments (a configuration). However, we shall not commit to any particular interpretation, and accordingly we shall use the word energy to designate it.

Using an energy-based account of probabilistic argumentation, we can use many concepts from statistical mechanics and information theory, such as entropy, and gain an intuitive understanding of the probabilistic setting. For example, the justification of such energy-based models can be found at the crossroad of information theory and statistical physics (see e.g. Jaynes, 1957), and we adapt it here for our argumentation setting.

Amongst possible configurations, it is a standard to accept the principle of insufficient reason (also called the principle of indifference): if there is no reason to prefer one configuration over alternatives, simply attribute the same probability to all of them. This principle dates back to Laplace when he was studying games of chance. A modern version of this principle is the principle of maximum entropy. In our setting, the entropy of a probability distribution of a set of labellings  $\mathcal{L} = \{l_1, l_2, \dots, l_n\}$  where the labelling  $l_i$  has probability  $P_i = P(l_i)$  is given as

$$S = - \sum_{l_i \in \mathcal{L}} P_i \cdot \ln(P_i). \quad (8)$$

The principle states that we should maximise the uncertainty we put in the probability distribution of labellings. If nothing more is known, then the principle of maximum entropy is satisfied by the distribution for which all the probabilities are equals.

However, the principle of insufficient reason has to be confronted to the principle of sufficient reason, which, in simple words, says that ‘nothing happens without a reason’. In our argumentative setting, the reason of the event of a labelling is fundamentally unknown, but such an event is nevertheless associated with an energy. The energy of a labelling  $l_i$  is denoted  $Q_i$ . This energy

may be interpreted as the (dis)credit put in a labelling (depending on the sign, as we can define an opposite energy  $Q' = -Q$ ): for example, it shall correspond to a measure of doubt or confidence in an epistemic setting, while it shall be interpreted as the (dis)utility of the labelling in a practical setting. In general, such energy shall have an interpretation depending on the considered context or application, but in any case, we assume that the average energy  $\sum_i P_i \cdot Q_i$  is known. From here, we cannot agree more with Jaynes (1957) stating that ‘in making inferences on the basis of partial information we must use that probability distribution which has maximum entropy to whatever is known’. On this basis, we have to maximize the entropy over all probability distributions with respect to the average energy and the total probability ( $\sum_i P_i = 1$ ). To maximize the entropy by taking into account these constraints, it is standard to use the method of Lagrange multipliers, and thus we consider the following Lagrangian expression:

$$L = S - \beta \sum_i P_i \cdot Q_i - \alpha \sum_i P_i. \quad (9)$$

Taking the derivative with respect to  $P_i$  gives

$$\ln P_i + \beta \cdot Q_i + \alpha = 0, \quad (10)$$

which yields

$$P_i = e^{-\alpha - \beta \cdot Q_i}. \quad (11)$$

We can then determine  $\alpha$  by recalling the total probability should be 1, and therefore we have

$$e^{-\alpha} \cdot \sum_i e^{-\beta \cdot Q_i} = 1. \quad (12)$$

So, the probability  $P_i$  of a labelling  $l_i$  is

$$P_i = \frac{e^{-\beta \cdot Q_i}}{\mathcal{Z}}, \quad (13)$$

where the normalizing factor  $\mathcal{Z}$  is called the partition function by analogy with physical systems:

$$\mathcal{Z} = \sum_i e^{-\beta \cdot Q_i}. \quad (14)$$

This distribution is also known as Gibbs or Boltzmann distribution and was historically first introduced in statistical mechanics to model gas. In counterpart systems studied in statistical physics, the constant  $\beta$  usually turns out to be inversely proportional to the temperature. In machine learning, this temperature is often used as a factor to ‘regularise’ the distribution over configurations or as parameter balancing the exploration and exploitation of alternative configurations, but for the sake of simplicity and because it is not necessary for our purposes, we omit it here for our argumentative framework (by setting it to 1).

We discussed in the previous section the factorisation of a distribution in terms of dependences amongst random labellings, but we did not catered much for the form that factors can take. In this regard and in light of the above theoretical insights, it is standard to put the factorised distribution introduced in Equation (2) under the form of a log-linear model, so we write factors as exponentials:

$$\phi_i(\mathbf{L}_i) = e^{-Q(\mathbf{L}_i)}, \quad (15)$$

where  $Q(\mathbf{L}_i)$  is called the energy function of the set  $\mathbf{L}_i$  of random labellings.

Factors are meant to break down a full joint distribution into manageable pieces, but we are eventually interested to regroup all these pieces to get results for the full joint distribution. So, using the exponential form of factors of Equation (15) into Equation (2), we obtain

$$P_{\Phi}(L_{A1}, L_{A2}, \dots, L_{An}) = \frac{e^{\sum_i -Q(L_i)}}{\mathcal{Z}_{\Phi}}. \quad (16)$$

If the arguments are assumed to be independent, we can associate one factor to every random labelling, and thus we obtain

$$P(L_{A1}, L_{A2}, \dots, L_{An}) = \frac{e^{-\sum_{A_i} Q(L_{A_i})}}{\mathcal{Z}_{\Phi}}, \quad (17)$$

which we can rewrite as

$$P(L_{A1}, L_{A2}, \dots, L_{An}) = \prod_{A_i} \frac{e^{-Q(L_{A_i})}}{\mathcal{Z}_{\Phi}}. \quad (18)$$

Each factor associated with an argument  $A$  can be interpreted as the probability of the {on, off}-labelling of the associated argument:

$$P(L_A = \text{on}) = \frac{e^{-Q(L_A=\text{on})}}{e^{-Q(L_A=\text{on})} + e^{-Q(L_A=\text{off})}}, \quad (19)$$

$$P(L_A = \text{off}) = \frac{e^{-Q(L_A=\text{off})}}{e^{-Q(L_A=\text{on})} + e^{-Q(L_A=\text{off})}}, \quad (20)$$

from which we retrieve Equation (1):

$$P(L_A = \text{off}) = 1 - P(L_A = \text{on}). \quad (21)$$

And thus the probability of a {on, off}-labelling  $l$  is

$$P(l) = \prod_{A \in \text{on}(l)} P(L_A = \text{on}) \prod_{A \in \text{off}(l)} (1 - P(L_A = \text{on})), \quad (22)$$

which is the setting of Li et al. (2011), and Fazzinga et al. (2013) for computing the probability of a labelling when arguments are probabilistically independent indeed.

Figuring out that a labelling  $l$  can be straightforwardly associated with an assignment  $\mathbf{I}$  to every random labelling, we can write that the probability of a labelling is exponentially proportional to the overall energy of this labelling:

$$P_{\Phi}(l) = \frac{e^{-Q(l)}}{\mathcal{Z}_{\Phi}}, \quad (23)$$

from which we retrieve Equation (13) when  $\beta = 1$ .

Using factorised probability distribution under the form of energy-based models, we have a direct connection with log-linear models of factor graphs and thus Markov networks as well as products of experts (see Section 6), so that well-established tools and techniques can be used for our present and future purposes.

In particular, when hidden variables are considered, we can write the factors in Equation (6) under the form of a log-linear model:

$$\phi_k(\mathbf{l}, \mathbf{h}) = e^{-Q_k(\mathbf{l}, \mathbf{h})}, \quad (24)$$

so that

$$P_\Phi(\mathbf{l}, \mathbf{h}) = \frac{e^{-Q(\mathbf{l}, \mathbf{h})}}{\mathcal{Z}_\Phi}, \quad (25)$$

where

$$Q(\mathbf{l}, \mathbf{h}) = - \sum_{k=1}^K Q_k(\mathbf{l}, \mathbf{h}). \quad (26)$$

By marginalising a labelling over the hidden variables, we may define  $P_\Phi(\mathbf{l})$  in terms of a quantity  $F(\mathbf{l})$  that we shall call the free energy of the labelling  $\mathbf{l}$  (we shall precise this quantity in the next sections):

$$P_\Phi(\mathbf{l}) = \frac{e^{-F(\mathbf{l})}}{\sum_{\mathbf{l}} e^{-F(\mathbf{l})}} \quad (27)$$

so that we retrieve the probability of a labelling as previously formulated in Equation (23) where hidden variables are considered. With the introduction of hidden variables with an energy-based model, we are now prepared to make a straightforward bridge with BMs which we will overview in the next section, towards argumentative BMs.

## 6. Boltzmann machines

The original BM (Ackley, Hinton, and Sejnowski, 1985) is a probabilistic undirected graphical model with binary stochastic units. The BM may be interpreted as a stochastic recurrent neural network and a generalisation of Hopfield networks (Hopfield, 1982) which feature deterministic rather than stochastic units. In contrast, however, to common neural networks, BMs are fully generative models, widely utilised for the purpose of feature extraction, dimensionality reduction and learning complex probability distributions.

The nodes of a BM are usually divided into observed (or visible) and latent (or hidden) units, where the former represent observations we wish to model and the latter capture the assumed underlying dependences among visible nodes. Postulating a BM of  $D$  visible and  $K$  hidden units, an assignment of binary values to the visible and hidden variables will be denoted as  $\mathbf{v} \in \{0, 1\}^D$  and  $\mathbf{h} \in \{0, 1\}^K$ , respectively. BMs are, in fact, special MRFs, which can be interpreted as PoE models in the case of RBMs (Hinton, 2002a). In a nutshell, a PoE models a probability distribution by combining the output from several simpler distributions (representing the experts). A PoE can be interpreted as a council of experts judging the veracity or the goodness of something (e.g. a sample) in an unanimous manner. In more details, a PoE features a joint probability distribution of the following general form:

$$P_{\text{PoE}}(\mathbf{v}, \mathbf{h}) = \frac{1}{\mathcal{Z}_{\text{PoE}}} \prod_{k=1}^K \phi_k(\mathbf{v}, \mathbf{h}), \quad (28)$$

where  $\phi_k(\mathbf{v}, \mathbf{h})$  are (non-linear) functions in general, not necessarily representing probability distributions and thus not necessarily normalised. The resulting joint distribution  $P_{\text{PoE}}(\cdot)$ , however, needs to be properly normalised by means of the normalisation constant denoted as  $\mathcal{Z}_{\text{PoE}}$  and

also referred to as the partition function. PoE contrast with so-called mixture of experts where the ‘beliefs’ of individual experts are averaged. The main characteristic of a PoE is that the product of beliefs can appear much sharper than the individual beliefs of each expert or their average. If the experts are chosen from the family of exponential function so that

$$\phi_k(\mathbf{v}, \mathbf{h}) = e^{-Q_k(\mathbf{v}, \mathbf{h})}, \quad (29)$$

then the resulting PoE model with exponential experts is an MRF represented by the following joint probability density:

$$P_{\text{MRF}}(\mathbf{v}, \mathbf{h}) = \frac{1}{Z_{\text{MRF}}} e^{-\sum_{k=1}^K Q_k(\mathbf{v}, \mathbf{h})}. \quad (30)$$

As alluded to when introducing the use of such distribution in the previous section, and in order to reflect their physical meaning, the sum of energies  $Q_k(\cdot)$ , denoted as  $Q(\cdot) = \sum_{k=1}^K Q_k(\cdot)$  is referred to as the system energy. In this view, every configuration of a machine is associated with an energy, and the most probable configuration is the configuration with the lowest energy.

A BM is a special case of a MRF, where the energy function is chosen to be linear with respect to its inputs and more specifically of the following form:

$$Q(\mathbf{v}, \mathbf{h}; \boldsymbol{\vartheta}) = -\mathbf{v}^\top \mathbf{L} \mathbf{v} - \mathbf{h}^\top \mathbf{J} \mathbf{h} - \mathbf{v}^\top \mathbf{W} \mathbf{h}, \quad (31)$$

where  $\mathbf{L}$ ,  $\mathbf{J}$  and  $\mathbf{W}$  are weight matrices representing the connection weights between visible-to-visible, hidden-to-hidden and visible-to-hidden units, respectively. Note that the diagonals of  $\mathbf{L}$  and  $\mathbf{J}$  are set to zero, as self-loops are disallowed in the model. Finally, as  $\boldsymbol{\vartheta} = \{\mathbf{L}, \mathbf{J}, \mathbf{W}\}$  we denote the set of all weight parameters.

Finally, an RBM is formed by dropping the connections between visible-to-visible and hidden-to-hidden units, by setting the corresponding transfer matrices  $\mathbf{L}$  and  $\mathbf{J}$  to zero. As a result, the joint model density can be formed as follows:

$$P_{\text{RBM}}(\mathbf{v}, \mathbf{h}; \boldsymbol{\vartheta}) = \frac{1}{Z_{\text{RBM}}} e^{-Q(\mathbf{v}, \mathbf{h}; \boldsymbol{\vartheta})} = \frac{1}{Z_{\text{RBM}}} e^{\mathbf{v}^\top \mathbf{W} \mathbf{h}}. \quad (32)$$

In an RBM, since the probability of generating a visible configuration is proportional to the product of the probabilities that the visible configuration would be generated by each of the hidden units acting alone, an RBM can be understood as a PoE with one expert per hidden unit, see Hinton (2002a). As we will see, this interpretation is quite appealing in the context of probabilistic argumentation, as one may interpret a hidden unit as an arguer giving some credits for the labelling of arguments.

If the marginal probability of a configuration is sought, then it is common to introduce the free energy of a visible vector  $\mathbf{v}$ , denoted  $-F(\mathbf{v}; \boldsymbol{\vartheta})$ :

$$e^{-F(\mathbf{v}; \boldsymbol{\vartheta})} = \sum_{\mathbf{h}} e^{-Q(\mathbf{v}, \mathbf{h}; \boldsymbol{\vartheta})}, \quad (33)$$

so that

$$P_{\text{RBM}}(\mathbf{v}; \boldsymbol{\vartheta}) = \frac{e^{-F(\mathbf{v}; \boldsymbol{\vartheta})}}{Z_{\text{RBM}}}. \quad (34)$$

Notice that the free energy can be computed in a time linear in the number of hidden units, see Hinton (2012).

When training a machine, the primary objective is to estimate the optimal model weights  $\boldsymbol{\vartheta}^*$  minimising the system’s energy  $Q(\cdot)$  for a given set of  $N$  observations denoted as  $\{\mathbf{v}_n\}_{n=1}^N$ .



RBM's are usually trained by means of maximum likelihood estimation (MLE); although alternative approaches do exist in the literature, they usually introduce a significant computational burden. By employing MLE, we aim at estimating the optimal model parameter configuration  $\boldsymbol{\vartheta}^*$ , which maximises the marginal log-likelihood for a given set of  $N$  observations  $\{\mathbf{v}_n\}_{n=1}^N$  which are assumed independent and identically distributed (i.i.d. – and thus the probability distribution of the  $N$  observations is the product of the probabilities for each observation):

$$\ln P_{\text{RBM}}(\{\mathbf{v}_n\}_{n=1}^N | \boldsymbol{\vartheta}) = \ln \prod_{n=1}^N P_{\text{RBM}}(\mathbf{v}_n | \boldsymbol{\vartheta}) = \sum_{n=1}^N \ln P_{\text{RBM}}(\mathbf{v}_n | \boldsymbol{\vartheta}). \quad (35)$$

From an alternative perspective, we aim at adjusting the model parameters so as to fit the true distribution of a training dataset. A measure of the discrepancy between the true data distribution, denoted  $P(\mathbf{v})$ , and the RBM density, denoted as  $P_{\text{RBM}}(\mathbf{v} | \boldsymbol{\vartheta})$ , is the Kullback–Leibler divergence (KL divergence), henceforth denoted as  $D_{\text{KL}}$ :

$$D_{\text{KL}} = \sum_{\mathbf{v}} P(\mathbf{v}) \ln \frac{P(\mathbf{v})}{P_{\text{RBM}}(\mathbf{v})} = \mathbb{E}_{P(\mathbf{v})}[\ln P(\mathbf{v})] - \mathbb{E}_{P(\mathbf{v})}[\ln P_{\text{RBM}}(\mathbf{v} | \boldsymbol{\vartheta})]. \quad (36)$$

The first expectation is basically the entropy of  $P(\mathbf{v})$  and it is thus constant with respect to the model parameters  $\boldsymbol{\vartheta}$ . Therefore the KL divergence is minimal when the second term is maximal. The second term is the expected log-likelihood which can be approximated as an average of the finite number of  $N$  observation as follows:  $\mathbb{E}_{P(\mathbf{v})}[\ln P_{\text{RBM}}(\mathbf{v})] \approx (1/N) \sum_{n=1}^N \ln P_{\text{RBM}}(\mathbf{v}_n)$ . In the asymptotic case, where  $N \rightarrow +\infty$ , this approximation tends to the exact expected log-likelihood. Therefore, in order to minimize the KL divergence between the true and model distributions, it is sufficient to maximise the log data likelihood. To do so, however, we first need to obtain the marginal log-likelihood  $P_{\text{RBM}}(\mathbf{v} | \boldsymbol{\vartheta})$  of the RBM by summing-out the hidden variables  $\mathbf{h}$  as follows:

$$\ln P_{\text{RBM}}(\mathbf{v} | \boldsymbol{\vartheta}) = \ln \sum_{\mathbf{h}} P_{\text{RBM}}(\mathbf{v}, \mathbf{h} | \boldsymbol{\vartheta}) \quad (37)$$

$$= \ln \sum_{\mathbf{h}} \frac{1}{\mathcal{Z}_{\text{RBM}}} e^{-Q(\mathbf{v}, \mathbf{h})} \quad (38)$$

$$= \ln \sum_{\mathbf{h}} e^{-Q(\mathbf{v}, \mathbf{h})} - \ln \sum_{\tilde{\mathbf{h}}} \sum_{\tilde{\mathbf{v}}} e^{-Q(\tilde{\mathbf{v}}, \tilde{\mathbf{h}})} \quad (39)$$

Note that the normalisation constant  $\mathcal{Z}_{\text{RBM}} = \sum_{\tilde{\mathbf{h}}} \sum_{\tilde{\mathbf{v}}} e^{-Q(\tilde{\mathbf{v}}, \tilde{\mathbf{h}})}$  is intractable and thus the joint distribution is not directly accessible to us. Maximizing the log-likelihood is usually achieved via gradient methods. Differentiate Equation (39) yields:

$$\frac{\partial \ln P_{\text{RBM}}(\mathbf{v} | \boldsymbol{\vartheta})}{\partial \boldsymbol{\vartheta}} = \mathbb{E}_{P_{\text{RBM}}(\mathbf{h} | \mathbf{v})} \left[ -\frac{\partial Q(\mathbf{v}, \mathbf{h})}{\partial \boldsymbol{\vartheta}} \right] - \mathbb{E}_{P_{\text{RBM}}(\tilde{\mathbf{v}}, \tilde{\mathbf{h}})} \left[ -\frac{\partial Q(\tilde{\mathbf{v}}, \tilde{\mathbf{h}})}{\partial \boldsymbol{\vartheta}} \right]. \quad (40)$$

The first term is the expectation of the probabilities of hidden units given an observed data-point under the current model of the RBM, and thus it does not pose a significant challenge. The second term is the expectation of the joint probability of visible and hidden units: as mentioned above, it is intractable as we are unable to directly normalise the joint distribution. For that reason, the gradient is most frequently approximated by means of contrastive divergence (CD)

(Hinton, 2002b, 2012), which has proven to be a fast and reasonably effective way to perform learning for RBMs.

CD is based on a simple principle, according to which it is sufficient to approximate the expectations in Equation (40), using only one value of  $\mathbf{v}$  and  $\mathbf{h}$ . More specifically, we employ the following approximations:

$$\mathbb{E}_{P_{\text{RBM}}(\mathbf{h}|\mathbf{v})} \left[ -\frac{\partial}{\partial \boldsymbol{\theta}} Q(\mathbf{v}, \mathbf{h}) \right] \cong -\frac{\partial}{\partial \boldsymbol{\theta}} Q(\mathbf{v}^0, \mathbf{h}^0), \quad (41)$$

$$\mathbb{E}_{P_{\text{RBM}}(\tilde{\mathbf{v}}, \tilde{\mathbf{h}})} \left[ -\frac{\partial}{\partial \boldsymbol{\theta}} Q(\tilde{\mathbf{v}}, \tilde{\mathbf{h}}) \right] \cong -\frac{\partial}{\partial \boldsymbol{\theta}} Q(\mathbf{v}^k, \mathbf{h}^k), \quad (42)$$

where  $(\mathbf{v}^0, \mathbf{h}^0)$  and  $(\mathbf{v}^k, \mathbf{h}^k)$  are computed according to the following process:

- (1) ‘Clamp’  $\mathbf{v}^0$  to an observed datapoint:

$$\mathbf{v}^0 = \mathbf{v}_n. \quad (43)$$

- (2) Draw  $\mathbf{h}^0$  from its conditional density:

$$\mathbf{h}^0 \sim P_{\text{RBM}}(\mathbf{h} | \mathbf{v}^0). \quad (44)$$

- (3) Obtain  $(\mathbf{v}^k, \mathbf{h}^k)$  after  $k$  Gibbs samples  $t = 1, \dots, k$  as follows:

$$\mathbf{v}^t \sim P_{\text{RBM}}(\mathbf{v} | \mathbf{h}^{t-1}), \quad (45)$$

$$\mathbf{h}^t \sim P_{\text{RBM}}(\mathbf{h} | \mathbf{v}^t). \quad (46)$$

CD- $k$  is applied by using the pair  $(\mathbf{v}^k, \mathbf{h}^k)$  as obtained by step 3 with  $k$  iterations. The conditional expectations necessary for the aforementioned draws are as follows:

$$P_{\text{RBM}}(v_i = 1 | \mathbf{h}) = \sigma(Q(v_i = 1)), \quad (47)$$

$$P_{\text{RBM}}(h_j = 1 | \mathbf{v}) = \sigma(Q(h_j = 1)), \quad (48)$$

where  $\sigma(x) = 1/(1 + e^{-x})$  is the sigmoid logistic function, and  $Q(v_i = 1)$  and  $Q(h_j = 1)$  are the energies of activation of a visible and hidden node, respectively, that is, the sum of the weights of connections from other active nodes:

$$Q(v_i = 1) = \mathbf{W}_i \mathbf{h}, \quad (49)$$

$$Q(h_j = 1) = \mathbf{v}^T \mathbf{W}_j, \quad (50)$$

where  $\mathbf{W}_i$  and  $\mathbf{W}_j$  denote the  $i$ th row and  $j$ th column of the weight matrix  $\mathbf{W}$ , respectively. Concluding the weight parameter updates are given as follows:

$$\Delta w_{ij} = \epsilon (v_i^0 h_j^0 - v_i^k h_j^k), \quad (51)$$

where  $\epsilon$  scales the change. In practice many tricks exist to improve the training (see the guide by Hinton (2012) for details), including adaptive learning rates and advanced gradient techniques such as Nesterov gradient, and the weights may be updated after a so-called epoch, i.e. after estimating the gradient on ‘mini-batches’ which are small sets of (typically 10 to 100) examples.

The use of the logistic function in Equation (47) can be generalized to the case of a unit with  $K$  alternative states (such a unit is called a softmax unit), in this case the probability to draw the unit in the state  $j$  is as follows:

$$P_j = \frac{e^{\mathcal{Q}_j}}{\sum_{i=1}^K e^{\mathcal{Q}_i}}. \quad (52)$$

A softmax unit can be seen as a set of binary units constrained so that only one of the  $K$  configurations occurs. Training for softmax units is identical to the training for standard binary units.

To generate samples from the learned model, Gibbs sampling is usually employed (Murphy, 2012), by iteratively sampling all random variables from their conditional probability densities. We also refer to that process as a ‘free run’. More specifically, we begin each free run from a random initial point  $(\mathbf{v}^0, \mathbf{h}^0)$  and progressively draw samples  $t = 1, 2, 3, \dots$  as follows:

$$v_i^t \sim P_{\text{RBM}}(v_i | \mathbf{h}^{t-1}), \quad \forall i \in [1, D], \quad (53)$$

$$h_j^t \sim P_{\text{RBM}}(h_j | \mathbf{v}^t), \quad \forall j \in [1, K]. \quad (54)$$

The exact conditionals for the Gibbs sampler are given in Equations (47) and (48). The bipartite structure of an RBM allows for the use of efficient block Gibbs sampling, where visible units are simultaneously sampled given fixed values of the hidden units, and hidden units are simultaneously sampled given the visible units.

As the samples for learning and the samples in the generative mode are similarly drawn, BMs offer the possibility to learn a compact model of a distribution while drawing samples from the model. In this perspective, instead of reinitialising the Markov chain to an observed data point after each weight update, we may initialise the Markov Chain at the state in which it ended after  $k$  iterations, also referred as persistent CD (Tieleman, 2008).

Whatever the learning procedure, the number of hidden units used in an RBM has a strong influence on its ability to learn distributions. Roux and Bengio (2008) showed that when the number of hidden nodes of an RBM is increased, there are weight values for the new nodes that guarantee improvement in the KL divergence between the data distribution and the model distribution. In particular, they showed that any distribution over a set  $\{0, 1\}^D$  can be approximated arbitrarily well (in the sense of the KL divergence) with an RBM with  $K + 1$  hidden nodes where  $K$  is the number of input vectors whose probability is not 0. In practice, the value of the number of hidden nodes is often chosen on the basis of experience (see Hinton, 2012 for a recipe) and the desired approximation.

## 7. Argumentative Boltzmann machines

To construct a BM that embodies abstract argumentation in our probabilistic setting, we reformulate argumentative labelling as a Gibbs sampling within the binary setting of BMs. Yet different machines can be built to perform the labelling of argumentation graphs in such setting. For this reason, we are in fact in front of different types of machines called *argumentative Boltzmann machines* in the sense that different machines can be proposed to label arguments. We are proposing two types here and we will focus on the second.

A first type of argumentative machine consists in a machine where the {on, off}-labelling of an argument is represented by a visible node. We call this machine a {on, off}-labelling 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 if, and

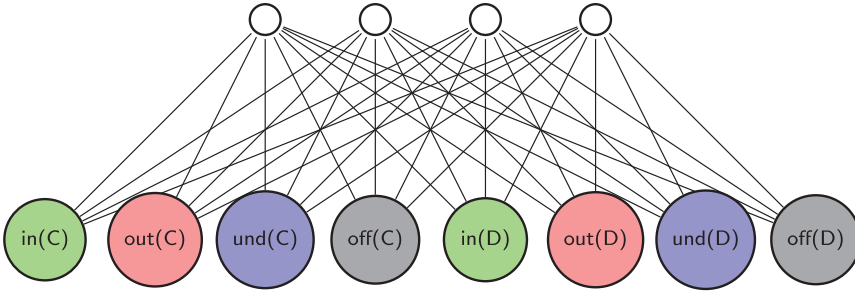


Figure 11. A  $\{\text{in, out, un, off}\}$ -labelling RBM for a hypothetical argumentation frame with two arguments C and D. Each visible node corresponds to a label of an argument.

only if, the corresponding node is switched off. When learning, any example clamped to the machine is a  $\{\text{on, off}\}$ -labelling (i.e. a sub-graph) to indicate whether an argument is **on** or **off**. So, a  $\{\text{on, off}\}$ -labelling machine shall learn a distribution of  $\{\text{on, off}\}$ -labellings and, when running in free mode, it shall sample  $\{\text{on, off}\}$ -labellings according to the learned distribution. Once a legal  $\{\text{on, off}\}$ -labelling is produced, one can compute the entailed grounded  $\{\text{in, out, un, off}\}$ -labelling using Algorithm 1. The advantage of such a machine is its simplicity: it boils down to a conventional BM chained with an algorithm to compute the grounded  $\{\text{in, out, un, off}\}$ -labelling of sampled graphs (or any other labelling based on the provision of such algorithm to compute this labelling). A major disadvantage is that this machine cannot discriminate **in**, **out** or **un** labellings (limiting the discriminative mode and possibly hindering possible useful mining of the network in search of relations amongst the status of arguments), and one cannot possibly guess the  $\{\text{in, out, un, off}\}$ -labelling of some arguments given the  $\{\text{in, out, un, off}\}$ -labelling of some other arguments.

To address the limitations of  $\{\text{on, off}\}$ -labelling BMs, we propose an alternative type of argumentative machines where a configuration of visible nodes represents a  $\{\text{in, out, un, off}\}$ -labelling. Such machines are henceforth referred to as a  $\{\text{in, out, un, off}\}$ -labelling machines. Different sorts of  $\{\text{in, out, un, off}\}$ -labelling machines are possible. The most instinctive way to construct such a machine is by pairing any label of every argument 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. An example of a  $\{\text{in, out, un, off}\}$ -labelling restricted Boltzmann machine is given in Figure 11.

An alternative  $\{\text{in, out, un, off}\}$ -labelling machine is a machine where each of the labels **in**, **out** and **un** is paired with one visible node, and only one. Then, an argument is labelled **off** when all the visible nodes of this argument are switched off (i.e. take the value 0). An example of this sort of  $\{\text{in, out, un, off}\}$ -labelling BM is given in Figure 12. The obvious advantage here is the reduction of the space complexity, and a priori the training time (since there are fewer weights to train).

As for notation, the energy of activation of an argument  $A$  being labelled **in** will be denoted  $Q(\text{in}(A))$ , and similarly  $Q(\text{out}(A))$  will denote the energy of  $A$  labelled **out**,  $Q(\text{un}(A))$  for  $A$  labelled **un**, and  $Q(\text{off}(A))$  for  $A$  labelled **off**. Since the four status of an argument is encoded by four different configurations of the associated units, we have the possibility to consider softmax units (see Section 6) for  $\{\text{in, out, un, off}\}$ -labelling BMs.

An assignment  $\mathbf{v}$  of the visible units corresponds to the assignment  $\mathbf{I}$  of the binary random labellings associated with the set  $\mathcal{A}_{\mathcal{G}}$  of arguments in  $\mathcal{G}$ , that is,  $\mathbf{v} = \mathbf{I}$ . Let  $\boldsymbol{\theta}$  be the parameters of the machine, and  $\mathbf{h}$  denote the assignment of the  $K$  binary hidden variables, we have the following probability distribution (to be compared with the joint probability distribution of an assignment

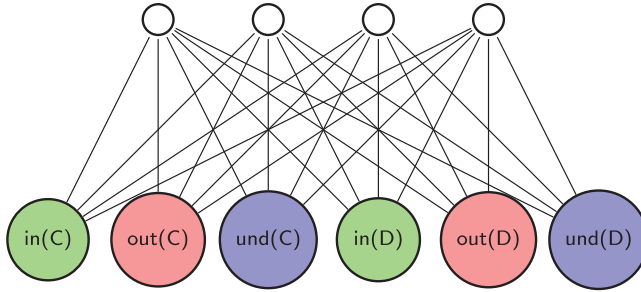


Figure 12. An alternative {in, out, un, off}-labelling RBM for a hypothetical argumentation frame with two arguments C and D.

of random labellings with hidden variables, see Equation (25)):

$$P(\mathbf{l}, \mathbf{h}; \boldsymbol{\vartheta}) = \frac{e^{-Q(\mathbf{l}, \mathbf{h}; \boldsymbol{\vartheta})}}{\mathcal{Z}_{\text{RBM}}}. \tag{55}$$

By marginalising a configuration of visible nodes (i.e. a labelling) over the hidden nodes in either {on, off}-labelling machines or in {in, out, un, off}-labelling machines, we can define  $P(\mathbf{v}; \theta)$  in terms of the free energy  $F(\mathbf{l}; \boldsymbol{\vartheta})$ , and thus we can obtain an interpretation of the energy of a labelling as the free energy of the corresponding configuration of visible nodes (to be compared with Equations (34) and (27)):

$$P(\mathbf{l}; \boldsymbol{\vartheta}) = \frac{e^{-F(\mathbf{l}; \boldsymbol{\vartheta})}}{\mathcal{Z}'_{\text{RBM}}}. \tag{56}$$

In this view, the energy of a labelling (see Definition 5) is the free energy of a configuration of visible nodes of a RBM. Remark that using Equation (34), we obtain the following ratio:

$$\frac{P(\mathbf{l}; \boldsymbol{\vartheta})}{P(\mathbf{l}'; \boldsymbol{\vartheta})} = e^{-F(\mathbf{l}; \boldsymbol{\vartheta}) + F(\mathbf{l}'; \boldsymbol{\vartheta})}. \tag{57}$$

Thus, we can exactly compute in linear time ratios between the probabilities of labellings, after a refined training of a RBM, or in cases where weights are set up by hand.

Focusing on grounded {in, out, un, off}-labelling machines, once a machine is well trained, the free energy of any illegal labelling shall be very high compared to legal labellings, in a manner such that the probability of any illegal labelling will be close to 0. So, a standard Gibbs sampling may sample illegal labellings with probability close to 0.

To ensure the production of grounded {in, out, un, off}-labellings, we may check ex post whether a produced labelling is grounded and discard it if not, but this solution may involve extra computation that may slow down the production of grounded labellings. So, we propose to tune Algorithm 1 for computing the grounded {in, out, un}-labelling of an argumentation graph into a probabilistic variant as given in Algorithm 2. This Algorithm 2 is called a ‘labelling random walk’ or more specifically a grounded {in, out, un, off}-labelling random walk in order to emphasise that first of all it features a random walk, on which any Gibbs sampling of a BM is based, and second that it is constrained by the construction of a grounded {in, out, un, off}-labelling. This walk shall be used to sample visible units so that we ensure that grounded {in, out, un, off}-labellings are sampled, and such a sampling is called a grounded {in, out, un, off}-sampling.

For the sake of the clarity of the presentation, we adopt the following notation. An argument A is labelled in with respect to a labelling  $l_i$  if, and only if,

- $A$  is not labelled in  $l_i$ , and
- $\forall B$  : if  $B$  attacks  $A$  then  $B \in \text{out}(l_i) \cup \text{off}(l_i)$ , and
- $A$  is drawn in:  $u \leq e^{Q(\text{in}(A))}/Z(A)$  where  $u$  is a random number in  $[0, 1]$  drawn from a uniform distribution, and

$$Z(A) = e^{Q(\text{in}(A))} + e^{Q(\text{out}(A))} + e^{Q(\text{un}(A))} + e^{Q(\text{off}(A))}.$$

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

An argument  $A$  is labelled out with respect of the labellings  $l_i$  and  $l_{i+1}$  if, and only if,

- $A$  is not labelled in  $l_i$ , and
- $\exists B$  :  $B$  attacks  $A$  and  $B \in \text{in}(l_{i+1})$ , and
- $A$  is drawn out:  $u \leq e^{Q(\text{out}(A))}/Z(A)$ .

We denote  $\text{OUT}(l_i, l_{i+1})$  the set of arguments eventually labelled out with respect to the labellings  $l_i$  and  $l_{i+1}$ . An argument  $A$  fulfilling the first two conditions but not drawn out is said ‘outable’, abbreviated  $\text{outable}(l_i, l_{i+1}, A)$ .

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.  $\text{inable}(l_i, A)$  or  $\text{outable}(l_i, l_{i+1}, A)$ .

We denote  $\text{OFF1}(l_i, l_{i+1})$  the set of arguments labelled off this way.

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:  $u \leq e^{Q(\text{off}(A))}/Z(A)$ .

We denote  $\text{OFF2}(l_{i+1})$  the set of arguments labelled off this way.

With this notation, we can now present a simple version of the grounded {in, out, un, off}-labelling random walk as given in Algorithm 2.

---

**Algorithm 2** Grounded {in, out, un, off}-labelling random walk.

---

- 1: **input** Machine parameters  $\mathcal{D}$ , argumentation frame  $\mathcal{G}$
  - 2:  $l_0 = (\emptyset, \emptyset, \emptyset, \emptyset)$
  - 3: **repeat**
  - 4:    $l'_0 \leftarrow l_i$
  - 5:   **repeat**
  - 6:      $\text{in}(l'_{i+1}) \leftarrow \text{in}(l'_i) \cup \text{IN}(l'_i)$
  - 7:      $\text{out}(l'_{i+1}) \leftarrow \text{out}(l'_i) \cup \text{OUT}(l'_i, l'_{i+1})$
  - 8:      $\text{off}(l'_{i+1}) \leftarrow \text{off}(l'_i) \cup \text{OFF1}(l'_i, l'_{i+1})$
  - 9:   **until**  $l'_i = l'_{i+1}$
  - 10:  $l_{i+1} \leftarrow l'_i$
  - 11:  $\text{off}(l_{i+1}) \leftarrow \text{off}(l_i) \cup \text{OFF2}(l_{i+1})$
  - 12: **until**  $l_i = l_{i+1}$
  - 13: **return**  $(\text{in}(l_i), \text{out}(l_i), A - \text{in}(l_i) \cup \text{out}(l_i) \cup \text{off}(l_i), \text{off}(l_i))$
-



The labelling random 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. The inner loop begins by labelling *in* all arguments not being attacked or whose attackers are *out* (line 6), and then it iteratively labels *out* any argument attacked by an argument labelled *in* (line 7). If an argument eligible for a label *in* or *out* is not labelled as such, then it is labelled *off* (line 8). The iteration of the inner loop continues until no more arguments can be labelled *in* or *out* (line 9). Any argument remaining unlabelled is then labelled *off* with its respective probability (line 11). If there is an argument labelled *off*, then the labelling proceeds in the inner loop. If there is no argument labelled *off*, then the walk terminates by finally labelling *un* all remaining arguments (line 13).

When learning, the machine will be shown a set of training examples where an example is a set of labelled arguments. To clamp the visible nodes of a machine with a set of labelled arguments, a visible node will be set at 1 if its labelling matches the example, otherwise it is set at 0. If a training example does not match any grounded  $\{\text{in, out, un, off}\}$ -labelling of the hypothetical framework, then it suggests that the hypothetical framework shall be changed and be adapted, but noisy settings may also occur. These issues are not addressed here. Notice that the learning procedure and the labelling walk are kept separated, thus we shall use the walk with any state-of-the-art learning algorithms or future improvements.

When labelling arguments with the labelling random walk, if the weight matrix  $\mathbf{L}$  is not set to 0 (e.g. in the case of a general BM), then the order in which the arguments are labelled matters. If the matrix  $\mathbf{L}$  is set to 0, as in the case of an RBM, then the order of labelling does not matter. Thus, a grounded  $\{\text{in, out, un, off}\}$ -labelling machine based on a conventional general BM exhibits a discrepancy between an ideal purely random walk amongst possible states of the machine (as suggested by a general BM setting) and the proposed labelling walk (Algorithm 2) meant to ensure grounded  $\{\text{in, out, un, off}\}$ -labelling. Indeed, this labelling walk is not purely random, as visible nodes are now considered in an order regimented by the construction of a grounded  $\{\text{in, out, un, off}\}$ -labelling. Thus, in order to integrate the argumentative constraints during all the walk, the RBM model appears more appropriate as the sampling of the visible nodes will be performed using the labelling random walk.

As an RBM can be interpreted as a PoE, we may interpret the labelling walk as the walk of a judge when given a coherent judgement concerning the status of arguments. At each step, the judge asks the advice of every expert (i.e. to every hidden node) to label the arguments, and this labelling occurs with respect to the constraints of a grounded labelling of the hypothetical argumentation frame. At the termination of the walk, the judge returns a labelling status to all the arguments of the frame.

Now, the hidden layer of BMs is often meant to extract features from input samples. The weights of each hidden unit represent those features, and define the probability that those features are present in a sample. For example, in face recognition, hidden units shall learn features such as edges at given orientations and positions, and these features shall be combined to detect facial features such as the nose, mouth or eyes. In argumentation, some sorts of labelling patterns may be captured by hidden nodes, and experiments in Section 8 will shed some lights on these considerations.

*Example 7.1* Let us illustrate the grounded  $\{\text{in, out, un, off}\}$ -labelling random walk. Suppose the argumentation frame in Figure 1. We may have the following walk:

- $l_0 = (\emptyset, \emptyset, \emptyset, \emptyset)$ ,
  - $l'_0 = (\emptyset, \emptyset, \emptyset, \emptyset)$ ,
    - suppose  $u \leq e^{Q(\text{in}(B))}/Z(B)$ , thus:  $\text{in}(l'_1) = \{B\}$ .

- suppose  $u > e^{Q(\text{out}(C))}/Z(C)$ , thus:  $\text{out}(l'_1) = \emptyset$ .
- $\text{un}(l'_1) = \emptyset$ .
- $\text{off}(l'_1) = \{C\}$ .
- $l'_1 = (\{B\}, \emptyset, \emptyset, \{C\})$ .
  - suppose  $u \leq e^{Q(\text{in}(D))}/Z(D)$ , thus:  $\text{in}(l'_2) = \{B, D\}$ .
  - $\text{out}(l'_2) = \emptyset$
  - $\text{un}(l'_2) = \emptyset$
  - $\text{off}(l'_2) = \{C\}$
- $l'_2 = (\{B, D\}, \emptyset, \emptyset, \{C\})$
- $l'_2 = l'_3$ .
- $l_1 = (\{B, D\}, \emptyset, \emptyset, \{C\})$ .
- $l_2 = l_1$

The resulting labelling is thus  $(\{B, D\}, \emptyset, \emptyset, \{C\})$ . Since the algorithm is non-deterministic, another walk may be:

- $l_0 = (\emptyset, \emptyset, \emptyset, \emptyset)$ ,
  - $l'_0 = (\emptyset, \emptyset, \emptyset, \emptyset)$ ,
    - suppose  $u \leq e^{Q(\text{in}(B))}/Z(B)$ , thus:  $\text{in}(l'_1) = \{B\}$ .
    - suppose  $u \leq e^{Q(\text{out}(C))}/Z(C)$ , thus:  $\text{out}(l'_1) = \{C\}$ .
    - $\text{un}(l'_1) = \emptyset$ .
    - $\text{off}(l'_1) = \emptyset$ .
  - $l'_1 = (\{B\}, \{C\}, \emptyset, \emptyset)$ .
    - suppose  $u \leq e^{Q(\text{in}(D))}/Z(D)$ , thus:  $\text{in}(l'_2) = \{B, D\}$ .
    - $\text{out}(l'_2) = \{C\}$
    - $\text{un}(l'_2) = \emptyset$
    - $\text{off}(l'_2) = \emptyset$
  - $l'_2 = (\{B, D\}, \{C\}, \emptyset, \emptyset)$
  - $l'_2 = l'_3$ .
- $l_1 = (\{B, D\}, \{C\}, \emptyset, \emptyset)$ .
- $l_2 = l_1$ .

So in this last walk, the labelling  $(\{B, D\}, \{C\}, \emptyset, \emptyset)$  was returned. In another run, when computing  $l'_0$  in the first time we may obtain  $l'_1 = (\emptyset, \emptyset, \emptyset, \{B\})$ , and the algorithm may return  $(\emptyset, \emptyset, \{C, D\}, \{B\})$ .

For any possible input, the grounded {in, out, un, off}-labelling random walk (algorithm 2) terminates, i.e. it exits by returning a labelling of all arguments of the argumentation frame. Indeed at each iteration of the outer loop, at least one argument is labelled until termination when all arguments get labelled.

**THEOREM 7.2 (Termination)** The grounded {in, out, un, off}-labelling random walk terminates.

*Proof* To prove the termination, we consider the sequence of pairs  $(A_0, l_0), (A_1, l_1), \dots, (A_n, l_n)$ , where  $A_i$  and  $l_i$  are the set of unlabelled arguments and the set of labelled arguments, that is, a partial labelling (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 terminates. ■

We say that a labelling algorithm is sound with respect to a type of labelling if, and only if, it returns a labelling of the argumentation frame as defined with respect to this type of labelling.

When the grounded {in, out, un, off}-labelling random walk terminates, it necessarily returns a grounded {in, out, un, off}-labelling of the argumentation frame, it is thus sound with respect to the grounded {in, out, un, off}-labelling.

**THEOREM 7.3 (Soundness)** The grounded {in, out, un, off}-labelling random walk is sound.

*Proof* To prove the soundness, we consider any terminated sequence  $(A_0, l_0), (A_1, l_1), \dots, (A_n, l_n)$ , and we show that  $l_n$  is a grounded {in, out, un, off}-labelling. To show that  $l_n$  is a grounded {in, out, un, off}-labelling, we consider the labelled argumentation sub-graph  $\mathcal{H}$  induced by the arguments labelled on and we show that this graph has a grounded {in, out, un}-labelling. To see that the graph  $\mathcal{H}$  has a grounded {in, out, un}-labelling  $l'$ , it is sufficient to observe that  $l'$  is a complete labelling, and  $\text{in}(l')$  is minimal (because no less arguments can be labelled in). Therefore,  $l_n$  is a grounded {in, out, un, off}-labelling. ■

Moreover, we say that a labelling algorithm is complete with respect to a type of labelling if, and only if, it can return any labelling of the argumentation frame as defined with respect to this type of labelling. The grounded {in, out, un, off}-labelling random walk can return any grounded {in, out, un, off}-labelling of the argumentation frame, it is thus complete.

**THEOREM 7.4 (Completeness)** The grounded {in, out, un, off}-labelling random walk is complete.

*Proof* To prove the completeness, we demonstrate that for any grounded {in, out, un, off}-labelling  $l$ , there exists a terminated sequence  $(A_0, l_0), (A_1, l_1), \dots, (A_n, l_n)$ , where  $l_n = l$ . For any terminal grounded {in, out, un, off}-labelling  $L$ , there exists a unique graph induced by the arguments labelled present 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, and thus any induced sub-graph. Consequently, for any grounded {in, out, un, off}-labelling  $l$ , there exists a terminated sequence  $(A_0, l_0), (A_1, l_1), \dots, (A_n, l_n)$  with  $l_n = l$ . ■

As mentioned earlier, once a machine is trained, the free energy of any illegal labelling shall be very high compared to legal labellings, in a manner such that the probability of any illegal labelling will be close to 0. So, a standard Gibbs sampling may sample illegal labellings with probability close to 0. Since the grounded {in, out, un, off}-labelling random walk is sound, we ensure that illegal labellings will not be sampled. And since the random walk is sound and complete, then a well trained grounded {in, out, un, off}-labelling machine coupled with a sample procedure like the random labelling walk shall produce a sound estimation of the probability distributions. Experiments in the next section will bring some evidences in this regard, within the boundaries of the experimented training procedures.

Concerning time complexity, the walk is a slight adaptation of Algorithm 1 for a probabilistic setting, meant to keep its polynomial complexity.

**THEOREM 7.5 (Time complexity)** Time complexity of the grounded {in, out, un, off}-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 at worst. 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 worse. Therefore, the time complexity of the grounded {in, out, un, off}-labelling random walk is polynomial. ■

Concerning space complexity, a major benefit of the proposed approach is to tackle the exponential space complexity of the sample space of a probabilistic argumentation frame (see Definition 4.1) by a compact machine whose the space complexity is reduced to weight matrices.

**THEOREM 7.6 (Space complexity)** Space complexity of the grounded {in, out, un, off}-labelling random walk is  $O(\max(|\mathbf{v}| \times |\mathbf{v}|, |\mathbf{h}| \times |\mathbf{h}|, |\mathbf{v}| \times |\mathbf{h}|))$ .

*Proof* A BM 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 11), 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 an RBM, we have  $\mathbf{L} = 0$  and  $\mathbf{J} = 0$ , therefore the space complexity is reduced to  $O(|\mathbf{v}| \cdot |\mathbf{h}|)$  in this case.

The labelling random walk is designed to ensure the production of grounded {in, out, un, off}-labellings. By sampling grounded {in, out, un, off}-labellings, we can then evaluate the probability of a labelling or the marginal probability of the labelling of some subset of arguments. The walk allows us to sample grounded {in, out, un, off}-labellings *on demand* (instead of checking ex post whether a sampled labelling is a grounded {in, out, un, off}-labelling and waiting that such a labelling eventually pops up), saving thus computational time when a valid labelling is requested. However, as BMs learn by mistakes too and since the random labelling walk may stuck the underlying Markov chain of the Gibbs sampling into some minima (corresponding to grounded {in, out, un, off}-labellings), one may seek a balance between the use of the random labelling walk and an ordinary sampling of visible units as given in Equation (47). This point will be experimentally explored by using an argumentative mixing rate in the next section.

## 8. Experiments

The experiments regard the use and effects of the grounded {in, out, un, off}-labelling random walk (Algorithm 2) in the learning mode and generative mode of an RBM.

To get experimental insights into the use of the grounded {in, out, un, off}-labelling random walk, we alternated grounded {in, out, un, off}-samplings and ordinary Gibbs samplings of conventional machines with an argumentative mixing rate  $p_m \in [0, 1]$ : we draw a random number  $u$  between  $[0, 1]$  from a uniform distribution, and if  $u \leq p_m$ , then an ordinary sampling is used to sample a configuration of visible nodes, otherwise a grounded {in, out, un, off}-sampling is performed. Hence, if the rate  $p_m = 0$ , then only grounded {in, out, un, off}-samplings are performed. If the rate  $p_m = 1$ , then we have a conventional RBM. This rate should not be confused with the mixing rate of the Markov chain, i.e. the rate of convergence to the Markov chain equilibrium distribution.

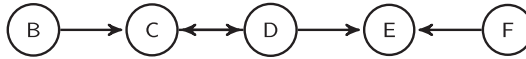


Figure 13. The hypothetical argumentation frame used for the first experiment.

The statistical distance between the empirical distribution  $P$  of the labellings drawn from a training dataset (approximating the source distribution) and the distribution  $P'$  learned by a machine was measured using the KL divergence (see Equation (36)) and the total variation distance (TV distance) denoted  $\delta(P, P')$  and defined as follows:

$$\delta(P, P') = \frac{1}{2} \sum_{l \in \mathcal{S}} |P(l) - P'(l)|, \quad (58)$$

where  $\mathcal{S}$  are the examples drawn from the training dataset.

The experiments concerned (i) a tiny hypothetical argumentation frame so that the source distribution, the distribution of reconstructed labellings (at the end of a training phase) and the distribution of generated labellings (at the end of a generative phase) could be extensively compared and that the weights could be examined (ii) a small hypothetical argumentation frame to test RBMs with different argumentative mixing rates and entropies of the training distributions (iii) larger frames to reconnoiter the limits of the tested machines. In all cases, the datasets were artificially generated (without any noise), so that this evaluation was not dependent of any particular domain. To build these datasets, each grounded {in, out, un, off}-labelling was associated a random energy and examples amongst labellings were drawn from the associated Gibbs-Boltzmann distribution.

### 8.1. First experiment

The first experiment regards the training of a grounded {in, out, un, off}-labelling machine based on a tiny hypothetical argumentation frame of 5 arguments, see Figure 13 – inducing  $4^5 = 1024$  different possible {in, out, un, off}-labellings, amongst which 32 grounded labellings.

The training consisted of 1000 passes over small batches, each containing 100 grounded {in, out, un, off}-labellings. The weights were updated after each pass, using adaptive learning rate (in the interval  $[0, 1]$ ) and Nestorov's accelerated gradient. The machine had 20 hidden nodes (the number of arguments' labels). The learning procedure was CD-4. The argumentative mixing rate was 0.5 (i.e. 50% of the generated samples were drawn using the grounded {in, out, un, off}-labelling random walk).

We present in Table 1 a comparison of the source distribution, the distribution of reconstructed labellings (taken from the 500th weight update to the last update of the training) and the distribution of generated labellings (at the end of a generative phase of the same duration than the CD-4 training phase). Numbers are rounded off.

As the table shows, the machine reconstructed the labellings in the training phase and generated the labellings in the generated phase according to a probability distribution closed to the source distribution.

Since the graphical model of a RBM can be interpreted as a PoE, where each hidden node is an expert, we can wonder whether this interpretation shall reflect in the weights learned during the training. For example, we can wonder whether an expert shall clearly favour a particular {in, out, un, off} labelling, or whether the illegal labelling of an argument shall be discarded by all the experts by setting a strong negative weight for this labelling.

To inspect whether we can retrieve some labelling patterns from the weights given by the experts after training, we display the weights learned by two experts in Figure 14. As the weights

Table 1. Comparing the empirical probability distribution of a source training set of labellings ( $P_{\text{source}}$  with an entropy 2.681) and the probability distribution over the same labellings but resulting from the training of a mixed RBM ( $P_{\text{training}}$ ) and the probability distribution over the same labellings but resulting from a generative period ( $P_{\text{gen}}$ ).

$P_{\text{source}}$	$P_{\text{training}}$	$P_{\text{gen}}$
0.158	0.157	0.161
0.13	0.13	0.127
0.129	0.127	0.125
0.106	0.103	0.104
0.071	0.071	0.068
0.071	0.07	0.069
0.048	0.046	0.043
0.047	0.046	0.046
0.047	0.045	0.047
0.039	0.039	0.041
0.039	0.037	0.038
0.021	0.021	0.023
0.012	0.013	0.012
0.012	0.012	0.012
0.01	0.01	0.01
0.008	0.007	0.006
0.008	0.008	0.007
0.005	0.007	0.006
0.005	0.006	0.007
0.005	0.005	0.005
0.005	0.005	0.003
0.004	0.006	0.01
0.004	0.006	0.007
0.004	0.005	0.004
0.002	0.005	0.007
0.002	0.002	0.005
0.002	0.004	0.003
0.002	0.002	0.002
0.001	0.001	0.002
0.001	0.002	0.001
0.001	0.002	0.001
0	0	0

Notes: The resulting KL divergence for the training period is 0.005 and the TV distance is 0.017. The resulting KL divergence for the generative period is 0.01 and the TV distance is 0.029.

show, the first expert did not appear to frankly back any particular labelling. The second expert seemed to favour labellings where the argument **B** is labelled as **off**, the argument **C** as **in** or to a less extent **un**, the argument **D** as **un** (which is incompatible with **C** labelled as **in**), and with no clear preference for the labelling of other arguments.

We conclude that a trained expert will not necessarily favour a particular labelling in any sharp manner, though some parts of labellings may be identified. As an analogy, we have a judge in front of a brouhaha of experts, and astonishingly enough, this judge will manage to label arguments by making the product of the expertises.

## 8.2. Second set of experiments

The second experiment regarded the performance of machines with respect to different entropies of the source distribution and for the hypothetical argumentation frame shown in Figure 15. With this frame,  $4^{10} = 104,8576$  labellings are possible, amongst which  $2^{10} = 1024$  are grounded {in, out, un, off}-labellings.

(a)					(b)				
	in	out	un	off		in	out	un	off
B	-0.164	-0.418	-0.408	-0.015	B	-6.989	-0.305	-0.296	6.786
C	-0.937	-0.505	-0.531	-0.119	C	3.411	-5.526	1.197	0.969
D	0.218	-0.419	-0.557	-0.665	D	-0.220	-0.292	1.244	-1.067
E	-0.778	-0.002	-2.536	0.414	E	-0.271	-0.102	-1.378	-0.188
F	0.571	-0.413	-0.395	-0.248	F	-0.347	-0.271	-0.276	0.145

Figure 14. Two trained experts, (a) and (b). Each table shows the weights given by an expert to the alternative labellings of arguments.

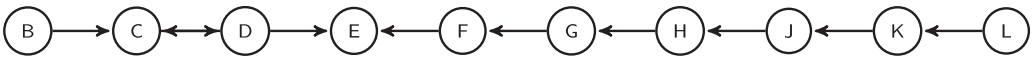


Figure 15. A hypothetical argumentation frame used for our second experiment.

The training consisted of 4000 passes over mini-batches, each containing 200 grounded {in, out, un, off}-labellings. The weights were updated after each pass, using adaptive learning rate (contained in the interval [0, 1]) and Nestorov’s accelerated gradient. The machine had 40 hidden nodes (the number of arguments’ labels). The learning procedure was CD-4.

We evaluated the machines by keeping track of the TV distance averaged over 100 different datasets, each dataset having a different entropy. Figure 16 shows, for different argumentative mixing rates, the tracked distances between (i) the source distribution and the distribution of reconstructed labellings (taken from the 2000th weight update to the last update of the training) and (ii) the source distribution and the distribution of generated labellings (at the end of a generative phase of the same duration than the CD-4 training phase from the 2000th weight update to the last update of the training). The KL divergence was not tracked because it may not be defined when some labellings are produced with a probability zero, especially when the learning is initiated. We show the runs for the cases where

- the mixing rate is 1 for the learning and the generative mode. This setting corresponds to a conventional RBM.
- the mixing rate is 0.5 for the learning and the generative mode. In this setting, visible nodes are sampled with the grounded {in, out, un, off}-labelling walk with a probability of 0.5. In the remainder, a machine with such a setting is called a balanced mixed argumentative machine.
- the mixing rate is 0 for the learning and the generative mode. In this setting, visible nodes are always sampled with the grounded {in, out, un, off}-labelling walk. In the remainder, a machine with such a setting is called a pure argumentative machine.
- the mixing rate is 1 for a first learning phase and 0 in a second learning phase, and then 0 for the generative mode.

We observe that a conventional machine exhibits a similar convergence than a balanced mixed machine. In comparison, a pure argumentative machine does not perform well. We conjecture that the reason holds in that pure argumentative machines ( $P_m = 0$ ) have no opportunity to learn from the production of invalid labellings because they necessarily produce grounded {in, out, un, off}-labellings, whereas other machines have this opportunity. This hypothesis is confirmed by the last setting where the performance of a pure argumentative machine is improved: in this setting, the machine has the opportunity to produce labellings which are not grounded and thus it has the opportunity to learn from these mistakes in a first learning phase. Nevertheless, in



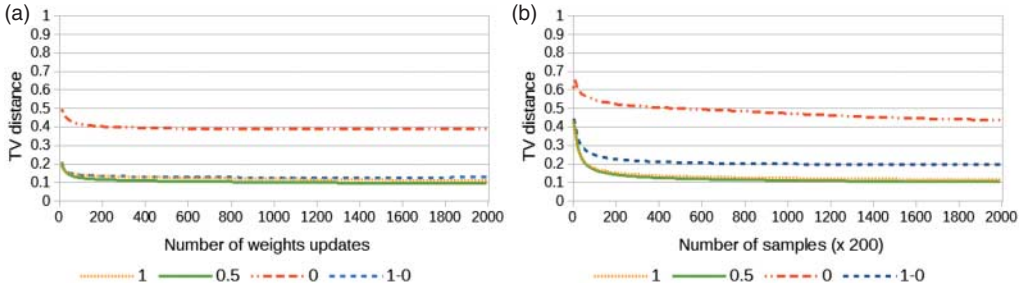


Figure 16. Progression of the TV distance for RBMs for different argumentative mixing rates. Results are averaged over 100 runs, each run learning a different dataset.

this last setting, and in comparison with the conventional and balanced mixed machines, a pure argumentative machine in the generative does not sample as well as in the training mode. We speculate here that the reason holds in that the sampling is hindered by the steady sampling from the grounded  $\{\text{in, out, un, off}\}$ -labelling walk, which tends to stuck the sampling of the Markov chain around some grounded  $\{\text{in, out, un, off}\}$ -labellings.

We give in Figure 17 the resulting statistical distances with respect to different entropies. The figures reveal a dispersion of the distances for distributions towards high entropies (up to a point where the statistical distances shall decrease for extreme entropies). The results also show that a balanced argumentative machine shall perform slightly better than a conventional machine. Of course such results hold for the experimental setting: one can better deal with high entropies by augmenting the number of hidden units (as we will see soon) or by simply increasing the training time for example. Moreover, we can expect that, in practice, argumentative routines shall bring distributions of labellings towards regions of low entropy.

Besides, the labelling walk is independent of the learning algorithm of a conventional BM, thus we should be able to train a machine as a conventional machine, use the labelling walk (in order to sample grounded  $\{\text{in, out, un, off}\}$ -labelling on demand) and obtain results in the generative mode similar to a conventional machine or a mixed machine. Figure 18 shows this case, and it confirms the hypothesis. It is an interesting result because it shows that a conventional learning algorithm shall be used for training, and labelling walks can be used to generate grounded labellings. Thus, one can take advantage of massive parallel computation offered by the graphical model of RBMs to speed up the training.

So far, we used  $\{\text{in, out, un, off}\}$ -grounded RBMs where any label of every argument is paired with one visible node (as illustrated in Figure 11). But as mentioned earlier, it is also possible to build more compact machines. For example, a natural alternative is to build  $\{\text{in, out, un, off}\}$ -grounded RBMs where each of the labels  $\text{in}$ ,  $\text{out}$  or  $\text{un}$  is paired with one visible node, and only one (as illustrated in Figure 12). Figure 19 shows the resulting statistical distances for this sort of machine. Comparing to the previous  $\{\text{in, out, un, off}\}$ -grounded RBMs, and along the same training setting, we observe similar results. As mentioned earlier, the advantage here is the reduction of the computational complexity (since there are fewer weights to train).

When overviewing BMs in Section 6, we mentioned that the number of hidden nodes used in an RBM influences its ability to learn distributions: as shown by Roux and Bengio (2008), when the number of hidden nodes of an RBM is increased, there are weight values for the new nodes that guarantee improvement in the KL divergence between the data distribution and the model distribution. This theoretical influence of the number of the hidden nodes on learning can be seen in Figure 20, where we can observed that a machine with 60 hidden nodes perform better than a machine with 15 nodes. These results also suggest that few hidden nodes shall be sufficient when

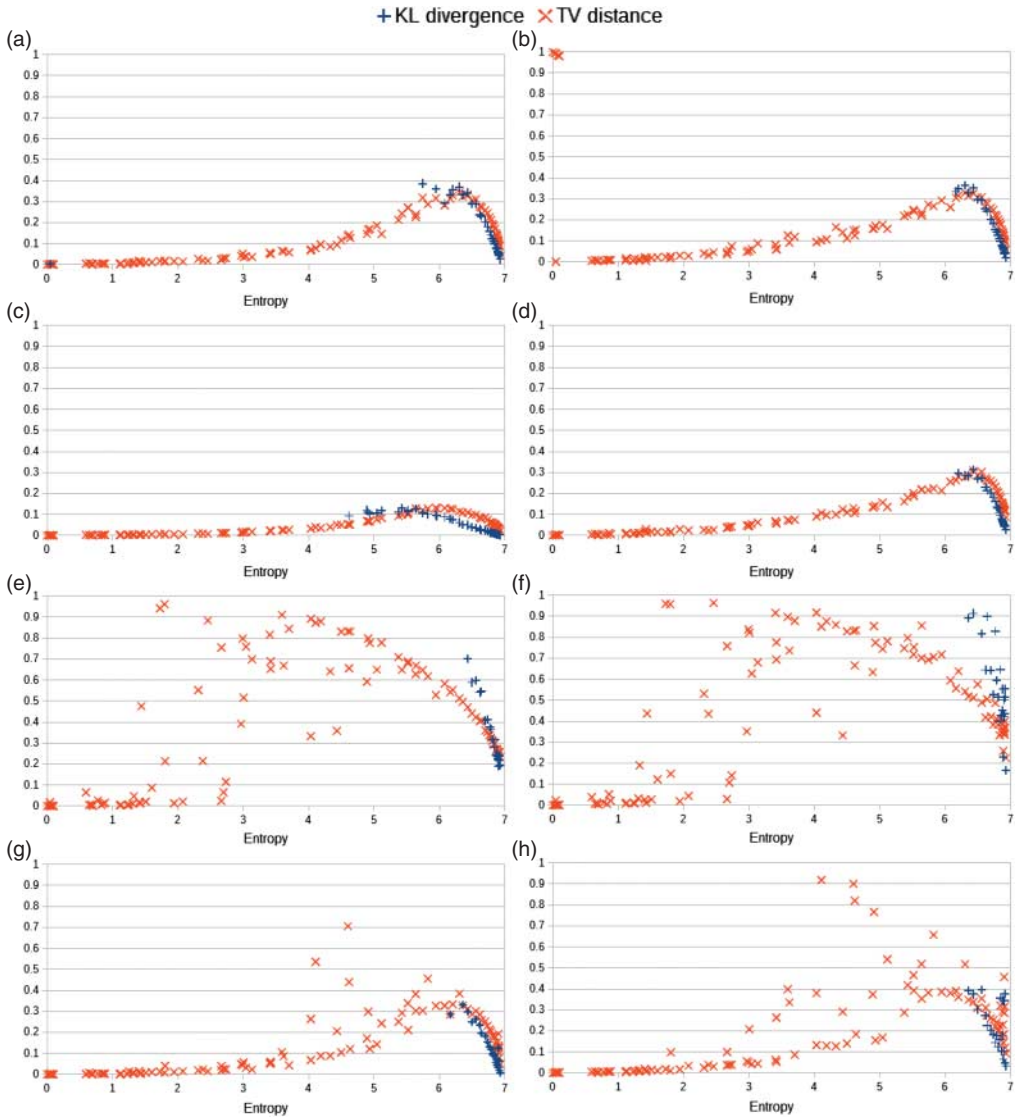


Figure 17. Statistical distances (KL divergence and TV distance) resulting after a training phase and a generative phase versus the entropy of the training probability distributions. (a) Learning mode with a mixing rate  $p_m = 1$  (conventional machine). (b) Generative mode with a mixing rate  $p_m = 1$ . (c) Learning mode with a mixing rate  $p_m = 0.5$ . (d) Generation mode with a mixing rate  $p_m = 0.5$ . (e) Learning mode with a mixing rate  $p_m = 0$  (only grounded labellings are produced). (f) Generative mode with a mixing rate  $p_m = 0$ . (g) Learning mode with a mixing rate  $p_m = 0$ , after a phase where the mixing rate  $p_m = 1$ . (h) Generative mode with a mixing rate  $p_m = 0$ .

modelling distributions with low entropies. When the entropy is 0, then the biases shall suffice to capture the considered labelling (which has a probability 1).

In light of these experiments, the labelling random walk appears thus as a mean to sample valid labellings *on demand*, without degraded performance with respect to a conventional RBM, and even with possible improvements. We show that a conventional learning algorithm

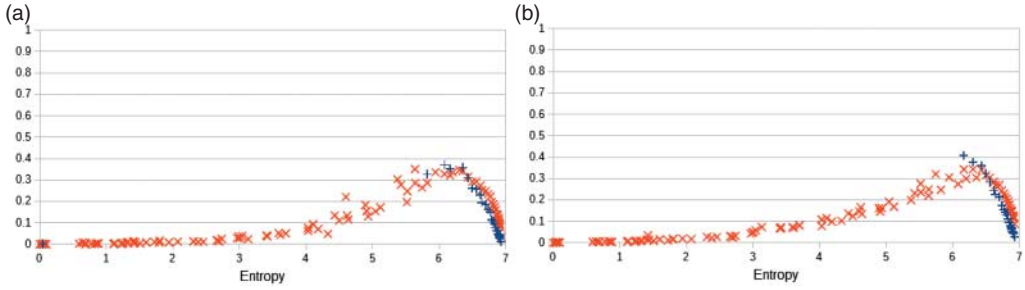


Figure 18. Statistical distances (KL divergence and TV distance) after a training phase and a generative phase versus the entropy of the training probability distributions. By definition of the experimental setting, the distances for the training mode are the distances of a conventional machine. (a) Learning mode with a mixing rate  $p_m = 1$  (conventional machine). (b) Generative mode with a mixing rate  $p_m = 0.5$ .

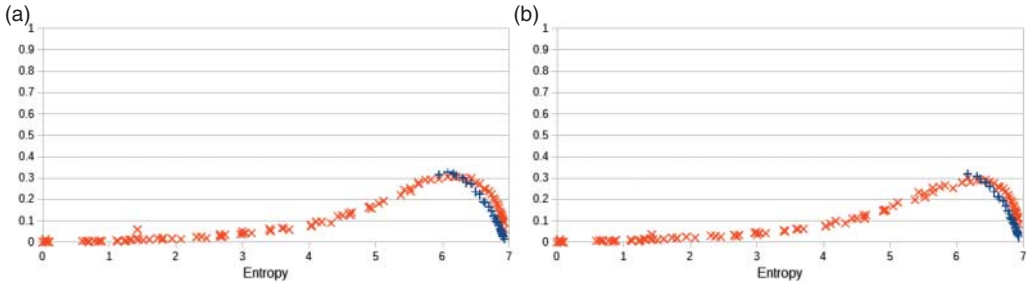


Figure 19. Statistical distances (KL divergence and TV distance) resulting after a training phase and a generative phase versus the entropy of the training probability distributions, for a {in, out, un, off}-labelling machine of the sort shown in Figure 12). (a) Learning mode with a mixing rate  $p_m = 0.5$ . (b) Generative mode with a mixing rate  $p_m = 0.5$ .

shall be used for training, and a labelling walk can be used to generate grounded labellings. Thus, one can take advantage of massive parallel computation offered by the graphical model of RBMs to speed up the training. Furthermore, since the labelling walk is well separated from the learning procedure, we can take advantages of variants of such procedures and future improvements. By increasing the number of sampled grounded {in, out, un, off}-labellings, we can sample more precise distributions from the machines (since the variance is automatically decreased with the number of samples). However, if grounded {in, out, un, off}-labellings are too often sampled using the labelling walk (as in the case of a pure argumentative machine), then the machine may not learn as it could from the production of mistaken labellings in the training mode, and the underlying Markov chain may not mix well in the generative mode.

### 8.3. Third set of experiments

We completed our experimental insights by extending the hypothetical argumentation frame with up to 20 arguments. We considered 10 frames, from 10 arguments to 20 arguments. In order to provide a compact and visual representation, we present each frame under the form of a submatrix of the adjacency matrix  $20 \times 20$  shown in Figure 21: the colour of the intersection at the  $i$ th row and  $j$ th column indicates whether the argument  $i$  does or not attack the argument  $j$ . If the intersection is white, then there is an attack, if the intersection is black, then there is no attack. So the first frame with 10 arguments is the upper left  $10 \times 10$  submatrix corresponding to the graph

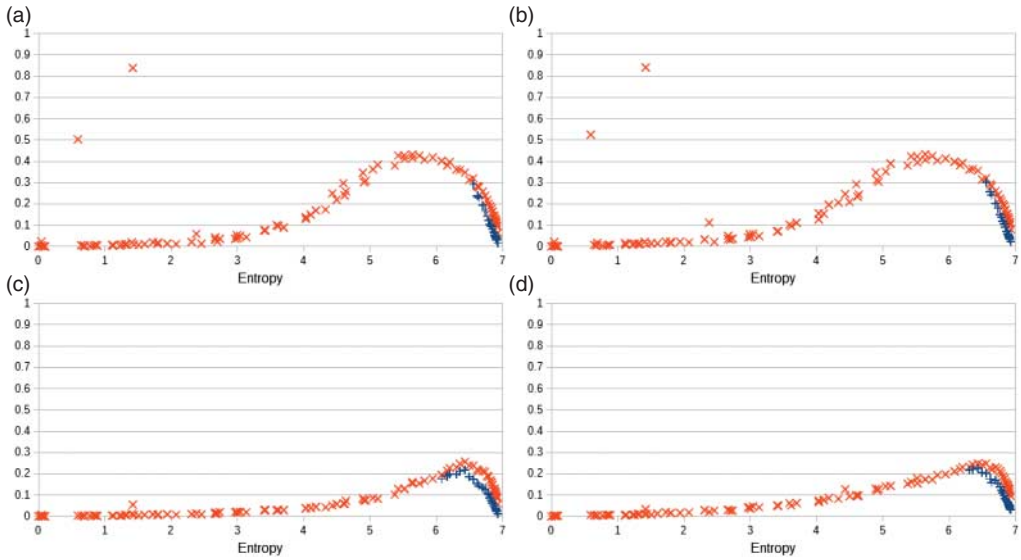


Figure 20. Statistical distances (KL divergence and TV distance) resulting after a training phase and a generative phase versus the entropy of the training probability distributions, for a {in, out, un, off}-labelling machine of the sort shown in Figure 12), with respect to settings with 15 hidden nodes (top graphs) and 60 hidden nodes (bottom graphs). (a) Learning mode with 15 hidden nodes. (b) Generative mode with 15 hidden nodes. (c) Learning mode with 60 hidden nodes. (d) Generative mode with 60 hidden nodes.

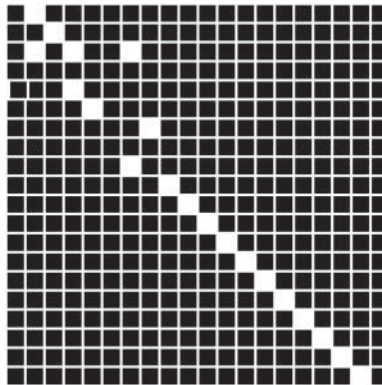


Figure 21. Adjacency matrix encoding argumentation graphs.

given in Figure 15. The second frame with 11 arguments is the  $11 \times 11$  submatrix containing the first frame, and so on. The largest frame with 20 arguments induces  $2^{20} = 1,048,576$  grounded {in, out, un, off}-labellings amongst  $4^{20} \approx 10^{12}$  possible labellings.

We kept the training setting of the second experiment for machines with an argumentative mixing rate 0.5, with the exception of the duration of the training and generative phases: in this experiment the training consisted of 6000 passes over mini-batches. We evaluated the machines by keeping track of the TV distances for four runs for the different frames and for two entropies of datasets, 3.466 and 5.198.

The resulting TV distances are given in Figure 22, for the two entropies 3.466 and 5.198, between (i) the source distribution and the distribution of reconstructed labellings (taken from

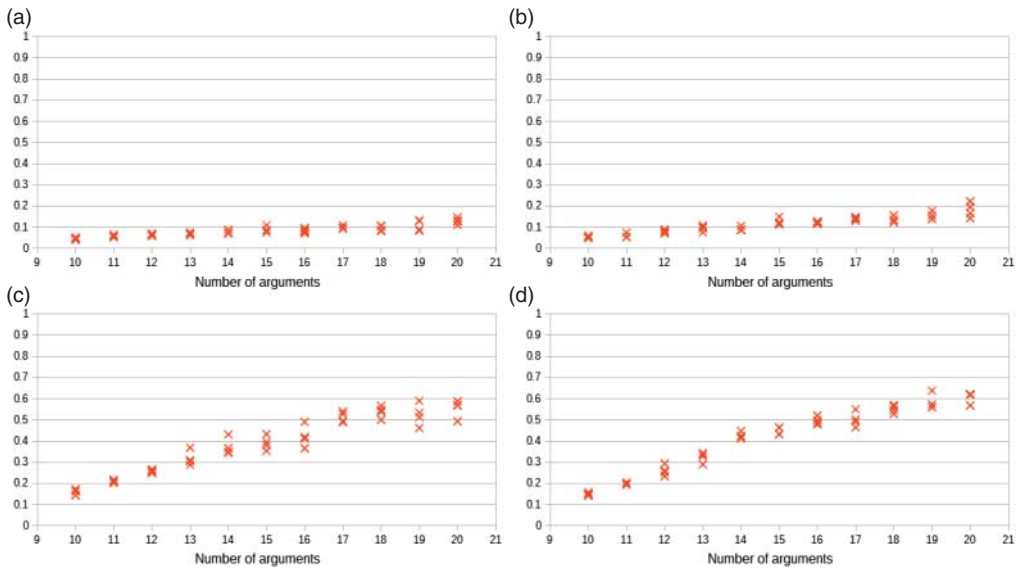


Figure 22. TV distances resulting after a learning phase and a generative phase (with a mixing rate  $p_m = 0.5$ ) versus the number of arguments. For any argumentation graph, the entropy was fixed to 3.466 (see (a) and (b)) and 5.198 (see (c) and (d)). (a) and (c) Learning mode. (b) and (d) Generative mode.

the 2000th weight update to the last update of the training) and (ii) the source distribution and the distribution of generated labellings (at the end of a generative phase of the same duration than the CD-4 training phase from the 2000th weight update to the last update of the training).

We observe that learning degrades when the number of arguments augments and when the entropy increases. Results shall improve with a longer training, or by considering more advanced learning settings such as persistent contrastive learning (Tieleman, 2008), stacked RBMs (i.e. deep BMs, see Salakhutdinov and Hinton, 2009) or some assumptions concerning the probabilistic dependences amongst arguments (as in the spirit of factor graphs). Such settings are left for future investigations.

## 9. Conclusion

We investigated an integration of abstract argumentation and probability theory with undirected graphical models and in particular with the graphical model of BMs. This combination allows us to relax assumptions on probabilistic independences amongst arguments, along a possible closed-form representation of a probabilistic distribution. We focused on generative learning, and we proposed a random labelling walk to generate grounded labellings ‘on demand’, while possibly keeping conventional training procedures. Thus, we can take advantage of massive parallel computation offered by the graphical model of RBMs to speed up the training. Since the labelling walk is well separated from the learning procedure, we can take advantages of variants of such learning procedures and future improvements. It also offers the possibility to learn on the fly a probability distribution of labelled arguments while producing labellings sampled according to the learned distribution.

Since the model of BMs is commonly interpreted as a model of neural networks, we are thus moving towards the construction of neuro-argumentative systems based on a principled account of probabilistic argumentation. Moreover, as the graphical model of RBMs can also be interpreted

as a PoE, our contribution can also be interpreted as a probabilistic model of an assembly of experts judging the status of arguments.

Future developments can be multiple, they range from instantiation of the abstract argumentation into specific frameworks (e.g. rule-based argumentation, see Riveret et al. (2015) for a setting which accounts for sub-arguments) to further benchmarking of variants of BMs such as deep machines for learning and inference in probabilistic argumentation. Different semantics may be addressed, and the modifications of hypothetical argumentation frames may be considered to account for unforeseen argumentative dependences. Towards structure learning, weights of argumentative machines shall be interpreted in order to extract the structure of hypothetical argumentation graphs. Finally, the utility of the approach has to be investigated by learning from labelled data in real applications instead of artificial distributions.

### Disclosure statement

No potential conflict of interest was reported by the authors.

### Funding

This work was supported by the Marie Curie Intra-European Fellowship PIEF-GA-2012-331472. Data61, formerly NICTA, is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

### References

- Ackley D. H., Hinton G. E., & Sejnowski T. J. (1985). A learning algorithm for Boltzmann machines\*. *Cognitive Science*, 9, 147–169.
- Baroni P., Caminada M., & Giacomin M. (2011). An introduction to argumentation semantics. *Knowledge Engineering Review*, 26, 365–410.
- Baroni P., Giacomin M., & Vicig P. (2014). *On rationality conditions for epistemic probabilities in abstract argumentation*. Proceedings of computational models of argument: Vol. 266. Frontiers in artificial intelligence and applications (pp. 121–132). IOS Press.
- Bondarenko A., Dung P. M., Kowalski R. A., & Toni F. (1997). An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93, 63–101.
- Cayrol C., & Lagasque-Schiex M.-C. (2013). Bipolarity in argumentation graphs: towards a better understanding. *International Journal of Approximate Reasoning*, 54, 876–899.
- d'Avila Garcez A. S., Gabbay D. M., & Lamb L. C. (2005). Value-based argumentation frameworks as neural-symbolic learning systems. *Journal of Logic and Computation*, 15, 1041–1058.
- d'Avila Garcez A. S., Gabbay D. M., & Lamb L. C. (2014). A neural cognitive model of argumentation with application to legal inference and decision making. *Journal of Applied Logic*, 12, 109–127.
- d'Avila Garcez A. S., Lamb L., & Gabbay D. (2008). *Neural-symbolic cognitive reasoning* (1st ed.). Springer.
- De Finetti B. (1974). *Theory of probability: a critical introductory treatment*. Wiley.
- Dondio P. (2014). Toward a computational analysis of probabilistic argumentation frameworks. *Cybernetics and Systems*, 45, 254–278.
- Dung P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77, 321–357.
- Dung P. M., & Thang P. M. (2010). Towards (Probabilistic) argumentation for jury-based dispute resolution. Proceedings of the 2010 conference on computational models of argument, COMMA'10 (pp. 171–182). IOS Press.
- Fazzinga B., Flesca S., & Parisi F. (2013). *On the complexity of probabilistic abstract argumentation*. IJCAI, IJCAI/AAAI.



- Fruhwirth T. (2009). *Constraint Handling Rules* (1st ed.). Cambridge University Press.
- Getoor L., Friedman N., Koller D., Pfeffer A., & Taskar B. (2007). Probabilistic relational models. *An introduction to statistical relational learning*. MIT Press.
- Getoor L., & Taskar B. (2007). *Introduction to statistical relational learning (Adaptive computation and machine learning)*. The MIT Press.
- Governatori G., Maher M. J., Billington D., & Antoniou G. (2004). Argumentation Semantics for Defeasible Logic. *Journal of Logic and Computation*, 14, 675–702.
- Governatori G., Rotolo A., & Sartor G. (2005). Temporalised normative positions in defeasible logic. In *Proceedings of the 10th International Conference on Artificial Intelligence and Law* (pp. 25–34). ACM.
- Hinton G. E. (2002a). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14, 1771–1800.
- Hinton G. E. (2002b). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14, 1771–1800.
- Hinton, G. (2012), “A Practical Guide to Training Restricted Boltzmann Machines,” Vol. 7700, *Springer Berlin Heidelberg*, pp. 599–619.
- Hinton A., Kwiatkowska M., Norman G., & Parker D. (2006). PRISM: A tool for automatic verification of probabilistic systems. *Proceedings of the 12th international conference on tools and algorithms for the construction and analysis of systems, TACAS’06* (pp. 441–444). Springer.
- Hopfield J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79, 2554–2558.
- Hunter A. (2013). A probabilistic approach to modelling uncertain logical arguments. *International Journal of Approximating Reasoning*, 54, 47–81.
- Hunter A., & Thimm M. (2014). Probabilistic argumentation with epistemic extensions and incomplete information. *CoRR*, abs/1405.3376.
- Jaynes E. T. (1957). Information theory and statistical mechanics. *Physical Review*, 106, 620–630.
- Koller D., & Friedman N. (2009). *Probabilistic graphical models: principles and techniques – adaptive computation and machine learning*. The MIT Press.
- LeCun Y., Chopra S., Hadsell R., Ranzato M., & Huang F. (2006). A tutorial on energy-based learning. In G. Bakir, T. Hofman, B. Schölkopf, A. Smola & B. Taskar (Eds.), *Predicting structured data*. MIT Press.
- Li S. Z. (1995). *Markov random field modeling in computer vision*. Springer.
- Li H. (2015). *Probabilistic argumentation*. University of Aberdeen.
- Li H., Oren N., & Norman T. J. (2011). *Probabilistic argumentation frameworks*. Revised selected papers of first international workshop theory and applications of formal argumentation, TAFA’11. *Lecture Notes in Computer Science* (pp. 1–16). Springer.
- Modgil S., & Caminada M. (2009). Proof theories and algorithms for abstract argumentation frameworks. *Argumentation in artificial intelligence*. Springer, pp. 105–129.
- Mozina M., Zabkar J., & Bratko I. (2007). Argument based machine learning. *Artificial Intelligence*, 171, 922–937.
- Murphy K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.
- Nute D. (2001). Defeasible logic. *Handbook of logic in artificial intelligence and logic programming*. Oxford University Press, pp. 353–395.
- Paolucci M., Kossmann D., Conte R., Lukowicz P., Argyrakis P., Blandford A., ... Helbing D. (2013). Towards a living earth simulator. *CoRR*, abs/1304.1903.
- Prakken H., & Sartor G. (1997). Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7.
- Rahwan I. (2008). Mass argumentation and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6, 29–37.
- Richardson M., & Domingos P. (2006). Markov logic networks. *Machine Learning*, 62, 107–136.
- Riveret R., Pitt J. V., Korkinof D., & Draief M. (2015). *Neuro-symbolic agents: Boltzmann machines and probabilistic abstract argumentation with sub-arguments*. *Proceedings of the 2015 international conference on autonomous agents and multiagent systems, AAMAS* (pp. 1481–1489).



- Riveret R., Prakken H., Rotolo A., & Sartor G. (2008). *Heuristics in argumentation: a game theory investigation*. Proceedings of computational models of argument, COMMA'08, Vol. 172 of Frontiers in artificial intelligence and applications (pp. 324–335). IOS Press.
- Riveret R., Rotolo A., & Sartor G. (2012a). *Norms and learning in probabilistic logic-based agents*. Proceedings of the 11th international conference in deontic logic in computer science, DEON'12: Vol. 7393. Lecture notes in computer science (pp. 123–138). Springer.
- Riveret R., Rotolo A., & Sartor G. (2012b). Probabilistic rule-based argumentation for norm-governed learning agents. *Artificial Intelligence and Law*, 20, 383–420.
- Riveret R., Rotolo A., Sartor G., Prakken H., & Roth B. (2007). *Success chances in argument games: a probabilistic approach to legal disputes*. Proceeding of the 20th conference on legal knowledge and information systems, JURIX'07 (pp. 99–108). IOS Press.
- Roth B., Riveret R., Rotolo A., & Governatori G. (2007). *Strategic argumentation: a game theoretical investigation*. Proceedings of the 11th international conference on artificial intelligence and law, ICAIL'07 (pp. 81–90). ACM.
- Roux N. L., & Bengio Y. (2008). Representational power of restricted Boltzmann machines and deep belief networks. *Neural Computation*, 20, 1631–1649.
- Salakhutdinov R., & Hinton G. E. (2009). Deep Boltzmann machines. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS: Vol. 5. JMLR Proceedings* (pp. 448–455).
- Sartor G. (2005). *Legal reasoning: a cognitive approach to the law*. Springer.
- Sato T. (2008). A glimpse of symbolic-statistical modeling by PRISM. *Journal of Intelligent Information Systems*, 31, 161–176.
- Schneider J., Groza T., & Passant A. (2013). A Review of Argumentation for the Social Semantic Web. *Semantic Web*, 4, 159–218.
- Sneyers J., Meert W., Vennekens J., Kameya Y., & Sato T. (2010). CHR(PRISM)-based probabilistic logic learning. *CoRR*, abs/1007.3858.
- Sneyers J., Schreye D. D., & Fruhwirth T. (2013). Probabilistic legal reasoning in CHRiSM. *Theory and Practice of Logic Programming*, 13, 769–781.
- Thimm M. (2012). *A probabilistic semantics for abstract argumentation*. Proceedings of the 20th European conference on artificial intelligence, ECAI'12, Vol. 242 of Frontiers in artificial intelligence and applications (pp. 750–755). IOS Press.
- Tieleman T. (2008). *Training restricted Boltzmann machines using approximations to the likelihood gradient*. Proceedings of the 25th international conference on machine learning (pp. 1064–1071). ACM.
- Torrioni P., Gavanelli M., & Chesani F. (2007). Argumentation in the semantic web. *IEEE Intelligent Systems*, 22, 66–74.
- Torrioni P., Gavanelli M., & Chesani F. (2009). Arguing on the semantic grid. *Argumentation in artificial intelligence*. Springer, pp. 423–441 (chap. 21).