Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200508

Learning Constraints for the Epistemic Graphs Approach to Argumentation

Anthony HUNTER
Department of Computer Science,
University College London, London, UK
(anthony.hunter@ucl.ac.uk)

Abstract. Epistemic graphs are a proposal for modelling how agents may have beliefs in arguments and how beliefs in some arguments may influence the beliefs in others. The beliefs in arguments are represented by probability distributions and influences between arguments are represented by logical constraints on these probability distributions. This allows for various kinds of influence to be represented including supporting, attacking, and mixed, and it allows for aggregation of influence to be captured, in a context-sensitive way. In this paper, we investigate methods for learning constraints, and thereby the nature of influences, from data. We evaluate our approach by showing that we can obtain constraints with reasonable quality from two publicly available studies.

Keywords. Probabilistic argumentation; Learning for argumentation; Non-normative argumentation.

1. Introduction

Argumentation often involves uncertainty. This can be uncertainty within an argument (e.g. uncertainty about the premises, or about the claim following the premises) or uncertainty between arguments (e.g. uncertainty about the nature of the support or attack by an argument on another). Further uncertainty arises when one agent is considering what arguments another agent believes (which can be important when the agent wants to persuade the other agent).

Following the results of an empirical study with participants [18], epistemic graphs have been introduced as a generalization of the epistemic approach to probabilistic argumentation [10,11]. In this approach, the graph is augmented with a set of epistemic constraints that can restrict the belief we have in an argument, and state how beliefs in arguments influence each other, with a varying degree of specificity. This is illustrated in Example 1.

Example 1. *Consider the graph in Figure 1, and let us assume that if* $\mathtt{D}$ *is strongly believed, and* $\mathtt{B}$ *or* $\mathtt{C}$ *is strongly disbelieved, then* $\mathtt{A}$ *is strongly believed, whereas if* $\mathtt{D}$ *is believed, and* $\mathtt{B}$ *or* $\mathtt{C}$ *is disbelieved, then* $\mathtt{A}$ *is believed. Furthermore, if* $\mathtt{B}$ *are* $\mathtt{C}$ *are believed, then* $\mathtt{A}$ *is disbelieved. These constraints could be reflected by the following formulae:* $\varphi_1 : p(\mathtt{D}) > 0.8 \wedge (p(\mathtt{B}) < 0.2 \vee p(\mathtt{C}) < 0.2) \rightarrow p(\mathtt{A}) > 0.8$*;*

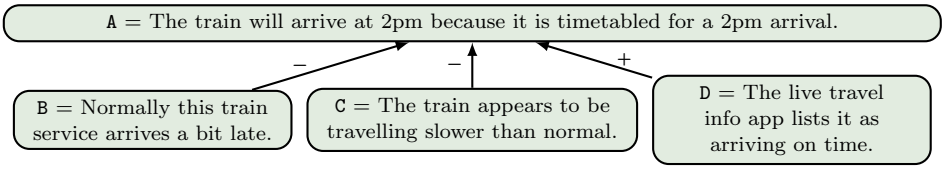

Figure 1. Example of an epistemic graph. The + (resp. -) label denote support (resp. attack) relations. These are specified via the constraints given in Example 1.

$\varphi_2 : p(\mathtt{D}) > 0.5 \wedge (p(\mathtt{B}) \leq 0.5 \vee p(\mathtt{C}) \leq 0.5) \rightarrow p(\mathtt{A}) > 0.5$*; and* $\varphi_3 : (p(\mathtt{B}) > 0.5 \wedge p(\mathtt{C}) > 0.5) \rightarrow p(\mathtt{A}) < 0.5$.

Epistemic graphs can model both attack and support as well as relations that are neither positive nor negative. The flexibility of this approach allows us to both model the rationale behind existing dialectical semantics (such by Dung [4]) and to completely deviate from them when required. The fact that we can specify the conditions under which arguments should be evaluated, and that we can include constraints between unrelated arguments, permits the framework to be more context–sensitive. It also allows for better modelling of imperfect agents, which can be important in multi–agent applications. Epistemic graphs are therefore a flexible and potentially valuable tool for argumentation, and [10] has already provided methods for harnessing epistemic graphs in user modelling for persuasion dialogues where knowing about what the other agent beliefs can help in strategically choosing arguments to present (see [8] for more on computational persuasion).

To date, it has been assumed that the constraints for an epistemic graph are available somehow, though no methods for acquiring have so far been proposed. Yet a key potential advantage of taking a probabilistic approach is that we can learn epistemic graphs from data. As a first step to realizing this potential, in this paper, we investigate methods for learning constraints, and thereby the nature of influences, from data. Our approach is a form of association rule learning [1] where the rules that we learn are in the form of probabilistic constraints (i.e. constraints for epistemic graphs). To evaluate our approach, we focus on publicly available data obtained in two published surveys: on the use of Wikipedia in higher education in Spain [16]; and on political attitudes in Italy [17]. These studies collected views about specific statements which can be regarded as arguments. Using our methods, we show that we can obtain constraints with reasonable quality (in terms of support and confidence).

In the rest of the paper, we present the following: (Section 2) Review of the definitions for epistemic graphs; (Section 3) Framework for learning epistemic constraints; (Section 4) Evaluation of framework with two datasets; (Section 5) Comparison with the literature; and (Section 6) Conclusion and discussion.

2. Restricted Epistemic Graphs

This section presents a simpler version of epistemic graphs than presented in [11]. Essentially, epistemic graphs are labelled directed graphs equipped with a

set of epistemic constraints for capturing the influences between arguments (as illustrated in Figure 1). Each node in the directed graph denotes an argument, and each arc denotes the influence of one argument on another. The label denotes the type of influence with options including positive (supporting), negative (attacking, and mixed. Both the labelled graph and the constraints provide information about the argumentation.

In this paper, we focus on the constraints rather than on the full power of the graphs. Let $\mathcal{G}$ denote a graph. Given the arguments in the graph, denoted $\mathsf{Nodes}(\mathcal{G})$, we consider a probability distribution $P : \wp(\mathsf{Nodes}(G)) \rightarrow [0,1]$ as being a probability assignment to each subset of the set of arguments such that this sums to 1 (i.e. $\sum_{X \subseteq \mathsf{Nodes}(G)} P(X) = 1$). The constraints restrict the set of probability distributions that satisfy the arguments (as we explain in the rest of this subsection).

Rather than consider any probability distribution in this paper, we will use finite probability distributions. For certain applications a restricted set of probability distributions can be used where the probability values come from a finite set of values [11]. This may be appropriate if we want to represent probability values as in a Likert scale [15]. It also has the benefit of always producing a finite set of distributions. However, for the approach to be coherent, this set should be closed under addition and subtraction (assuming the resulting value is in the $[0,1]$ interval) and should contain 1.

Definition 1. *A finite set of rational numbers from the unit interval* Π *is a* **restricted value set** *iff* $1 \in \Pi$ *and for any* $x, y \in \Pi$ *it holds that if* $x + y \leq 1$*, then* $x + y \in \Pi$*, and if* $x - y \geq 0$*, then* $x - y \in \Pi$.

Since we will only consider restricted value sets, we will refer to them as value sets. Examples include $\{0,1\}$, $\{0,0.5,1\}$, and $\{0,0.25,0.5,0.75,1\}$.

A probability distribution P for a value set Π is a probability distribution such that for each $\Gamma \subseteq \mathsf{Nodes}(G)$, $P(\Gamma) \in \Pi$. We will assume that all our probability distributions are with respect to a given value set. We denote the set of all belief distributions on $\mathsf{Nodes}(\mathcal{G})$ by $\mathsf{Dist}(\mathcal{G})$, and the set of restricted distributions for value set Π by $\mathsf{Dist}(\mathcal{G}, \Pi)$

Based on a given graph and restricted value set, we can now define the epistemic language. In this paper, we will only consider a sublanguage of that defined in [11].

Definition 2. *The* **restricted epistemic language** *based on graph* $\mathcal{G}$ *and a restricted value set* Π *is defined as follows: an* **epistemic atom** *is of the form* $P(\alpha)\#x$ *where* $\# \in \{<, \leq, =, \geq, >\}$*,* $x \in \Pi$ *and* $\alpha \in \mathsf{Nodes}(\mathcal{G})$*; an* **epistemic formula** *is a Boolean combination of epistemic atoms.*

Example 2. *Let* $\Pi = \{0, 0.5, 1\}$*. In the restricted epistemic language w.r.t.* Π*, we can only have atoms of the form* $p(\alpha)\#0$*,* $p(\alpha)\#0.5$*, and* $p(\alpha)\#1$*, where* $\alpha \in \mathsf{Nodes}(\mathcal{G})$ *and* $\# \in \{<, \leq, =, \geq, >\}$*. From these atoms we compose epistemic formulae, using the Boolean connectives, such as* $p(\alpha) \leq 0.5 \rightarrow \neg(p(\beta) \geq 0.5)$.

The semantics for constraints come from probability distributions $P \in \mathsf{Dist}(\mathcal{G}, \Pi)$, which assign probabilities to sets of arguments. Each $\Gamma \subseteq \mathsf{Nodes}(\mathcal{G})$ corresponds to a possible world where the arguments in Γ are true.

Definition 3. *The* **probability of an argument** *is defined as the sum of the probabilities of the worlds containing it:* $P(\alpha) = \sum_{\Gamma \subseteq \mathsf{Nodes}(\mathcal{G}) \text{ s.t. } \alpha \in \Gamma} P(\Gamma)$.

We say that an agent believes an argument α to be acceptable to some degree if $P(\alpha) > 0.5$, disbelieves α to be acceptable to some degree if $P(\alpha) < 0.5$, and neither believes nor disbelieves α to be acceptable when $P(\alpha) = 0.5$. Using this, we can finally produce (restricted) satisfying distributions of an epistemic atom, and therefore of an epistemic formula:

Definition 4. *Let* Π *be a value set and let* $p(\alpha)\#v$ *be an epistemic atom where* $\# \in \{<, \leq, =, \geq, >\}$. *The* **satisfying distributions**, *or equivalently* **models**, *of* $p(\alpha)\#v$ *are defined as* $\mathsf{Sat}(p(\alpha)\#v) = \{P' \in \mathsf{Dist}(\mathcal{G}) \mid P'(\alpha)\#v\}$. *The* **restricted satisfying distribution** *of* $\psi = p(\alpha)\#v$ *w.r.t.* Π *are defined as* $\mathsf{Sat}(\psi, \Pi) = \mathsf{Sat}(\psi) \cap \mathsf{Dist}(\mathcal{G}, \Pi)$.

The set of satisfying distributions for a given epistemic formula is as follows where ϕ and ψ are epistemic formulae: $\mathsf{Sat}(\phi \wedge \psi) = \mathsf{Sat}(\phi) \cap \mathsf{Sat}(\psi)$; $\mathsf{Sat}(\phi \vee \psi) = \mathsf{Sat}(\phi) \cup \mathsf{Sat}(\psi)$; and $\mathsf{Sat}(\neg\phi) = \mathsf{Sat}(\top) \setminus \mathsf{Sat}(\phi)$. For a set of epistemic formulae $\Phi = \{\phi_1, \ldots, \phi_n\}$, the set of satisfying distributions is $\mathsf{Sat}(\Phi) = \mathsf{Sat}(\phi_1) \cap \ldots \cap \mathsf{Sat}(\phi_n)$. The same holds when restricting probabilities to a value set Π.

Example 3. *Consider the formula* $p(\mathtt{A}) > 0.5 \rightarrow \neg(p(\mathtt{B}) > 0.5)$ *with* $\Pi = \{0, 0.5, 1\}$. *Examples of probability distributions that satisfy the formula include* P_1 *s.t.* $P_1(\varnothing) = 1$, P_2 *s.t.* $P_2(\varnothing) = P_2(\{\mathtt{A}\}) = 0.5$, P_3 *s.t.* $P_3(\{\mathtt{A}\}) = 1$, *or* P_4 *s.t.* $P_4(\{\mathtt{A}\}) = P_3(\{\mathtt{A}, \mathtt{B}\}) = 0.5$ *(omitted sets are assigned* 0*). The probability distribution* P_5 *s.t.* $P_5(\{\mathtt{A}, \mathtt{B}\}) = 1$ *does not satisfy the formula.*

The restricted epistemic language does not incorporate features of the full epistemic language (as presented in [11]) such as terms that are Boolean combinations of arguments (e.g. $P(\mathtt{B} \vee \mathtt{C}) > 0.6$ which says that the probability argument B or argument C is greater than 0.6) or summation of probability values (such as $P(\mathtt{A}) + P(\mathtt{B}) \leq 1$ which says that the sum of probability A and probability B is less than or equal to 1). Nonetheless, the restricted epistemic language is a useful sublanguage as a starting point for learning constraints. We focus on this sublanguage in this paper as it simplifies the presentation and evaluation.

3. Learning Framework

We now present a general framework for generating a class of epistemic constraints from data as follows: we define the format for the data, the format for constraints that we will learn, and an algorithm for learning these constraints. To illustrate, we will use examples taken from two studies (that we will discuss further in Section 4) concerning use of Wikipedia in higher education in Spain [16], and political attitudes in Italy [17].

3.1. Input for Learning

We assume that each item of data is a function that gives a value on an 11 point scale to each attribute. We use a data item to represent the responses that a

	Pu3	Qu1	Qu3	Enj1
903	0.3	0.5	0.3	0.7
904	0.9	0.9	0.9	0.9
905	0.7	0.7	0.5	0.9
908	0.3	0.5	0.7	0.3
909	0.5	0.5	0.7	0.7

Table 1. Some rows and columns of data from the Spanish study (after mapping Likert values to our 11 point scale) where Pu3 denotes the argument that "Wikipedia is useful for teaching", Qu1 denotes the argument that "Articles in Wikipedia are reliable", Qu3 denotes the argument that "Articles in Wikipedia are comprehensive", and Enj1 denotes the argument that "Articles in Wikipedia stimulate curiosity".

participant gives to each question where the attribute denotes the statement (i.e. the argument), and the assignment is their answer (as illustrated in Table 1).

Definition 5. *A* **data item** *is a function d from a set of attributes to a set of values. A* **dataset**, $D = \{d_1, \ldots, d_n\}$, *is a set of data items over attributes (i.e. arguments) $A = \{a_1, \ldots, a_m\}$. So for $d \in D$, and for $\alpha \in A$, $d(\alpha) \in \{0, 0.1, 0.2, , \ldots, 0.9, 1\}$.*

The Spanish and Italian studies used Likert scales (7, 8 and 10 point scales) for recording participants responses to arguments. For a common format, we map each value in the Likert scale to our 11 point scale (e.g. for the 7 point scale, we use the mapping $1 \mapsto 0$, $2 \mapsto 0.2$, $3 \mapsto 0.3$, $4 \mapsto 0.5$, $5 \mapsto 0.7$, $6 \mapsto 0.8$, and $7 \mapsto 1$). So we use each answer in the Likert scale as a proxy for the participant's belief in the argument. The 11-point scale allows us to represent total disbelief (i.e. 0), total belief (i.e. 1), and the values in between obtainable with 0.1 graduations.

Example 4. *Consider Table 1. From row 903, we get $d_{903}(\texttt{Pu3}) = 0.3$, $d_{903}(\texttt{Qu1}) = 0.5$, $d_{903}(\texttt{Qu3}) = 0.3$, and $d_{903}(\texttt{Enj1}) = 0.7$.*

Given the set of arguments in the data, we then identify relationships between them. For a pair of arguments α and β, we say that α **influences** β if a change in the belief in α will potentially result in the change in the belief in β. For instance, an argument influences another argument if it appears to attack it (i.e. it could be regarded as a counterargument), or if it appears to support it. But relationships may be more subtle or mixed (see [11] for more details).

Definition 6. *An* **influence tuple** *is a tuple $(\{\alpha_1, \ldots, \alpha_n\}, \beta)$, where $\{\alpha_1, \ldots, \alpha_n\} \subseteq \mathsf{Nodes}(\mathcal{G}) \setminus \{\beta\}$ and $\beta \in \mathsf{Nodes}(\mathcal{G})$ and each α_i influences β. We refer to each α_i as an* **influencer** *and β as an* **influence target**.

In this paper, we identified influence tuples by hand (i.e. by reading the statements in order to judge which arguments might be influenced by each argument). Potential alternatives to doing this by hand include automated reasoning with background knowledge about the arguments (such as causal relationships), and natural language processing to find logical relationships such as attack.

Example 5. *Consider the arguments* Dw1 *to* Dw3 *in Figure 2. By inspection we may regard* Dw2 *and* Dw3 *as attackers of* Dw1*, and so treat* Dw2 *and* Dw3 *as influencers of* dw1*. Hence, the influence tuple is* $(\{\texttt{Dw2}, \texttt{Dw3}\}, \texttt{Dw1})$.

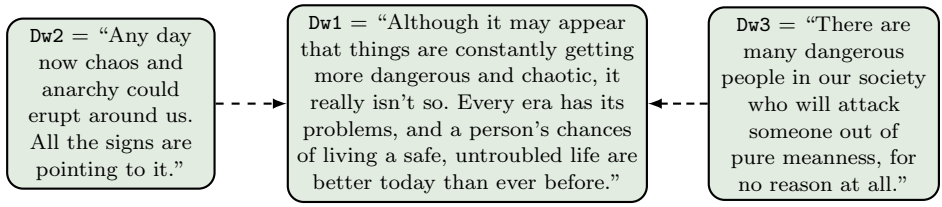

Figure 2. Arguments from the Italian study considered in Example 5. The dashed arcs denote influences.

So the input to the induction process is a data tuple and a set of influence tuples which provides extra information to guide the learning process. The learning process will ascertain (for the population of the study) whether there is indeed a relationship between some/all of the influencers and the influence target and if so, what the nature of that influence is. For instance, it could be that one argument does indeed contradict another argument, but for the population of a study, most people believe the attacker and the attackee. In this way, we want constraints that represent the beliefs of the population of the study rather than represent some normative interpretation of the arguments.

3.2. Output from Learning

The aim of learning is to take the input (a data set and a set of influence tuples) and return a set of constraints where each constraint is a rule. This set of rules will be a subset of the candidate rules defined next. Obviously each candidate rule is an epistemic formula (according to Definition 2).

Definition 7. *Let* $I = (\{\beta_1, \ldots, \beta_n\}, \alpha)$ *be an influence tuple, and* Π *be a value set. The set of* **candidate rules** *for* I *and* Π *is*

$$\begin{aligned}\mathsf{Rules}(I,\Pi) = \{&p(\gamma_1)\#_1 v_1 \wedge \ldots \wedge p(\gamma_k)\#_k v_k \rightarrow p(\alpha)\#_{k+1} v_{k+1} \mid \\ &\{\gamma_1, \ldots, \gamma_k\} \subseteq \{\beta_1, \ldots, \beta_n\} \text{ and } \#_i \in \{\leq, >\} \text{ and } v_i \in \Pi \setminus \{0,1\}\}\end{aligned}$$

Example 6. *Let* $I = (\{\texttt{Qu1}\}, \texttt{Enj1})$ *be an influence tuple and let* $\Pi = \{0, 0.5, 1\}$. *From this, the set of candidate rules* $\mathsf{Rules}(I, \Pi)$ *is*

$$\begin{array}{ll} p(\texttt{Qu1}) > 0.5 \rightarrow p(\texttt{Enj1}) > 0.5 & p(\texttt{Qu1}) > 0.5 \rightarrow p(\texttt{Enj1}) \leq 0.5 \\ P(\texttt{Qu1}) \leq 0.5 \rightarrow p(\texttt{Enj1}) > 0.5 & p(\texttt{Qu1}) \leq 0.5 \rightarrow p(\texttt{Enj1}) \leq 0.5 \end{array}$$

So for each influence tuple, the output of the induction process will be a set of rules, and these will be selected from the candidates in $\mathsf{Rules}(I, \Pi)$.

3.3. Generate Rules from Data

In the following, we introduce the 2-way generalization step that generates a rule from a data item. It has a precondition (above the line) and a postcondition (below the line). In the postcondition, the epistemic atoms in the rule are either of the form greater than 0.5 or less than or equal to 0.5 (i.e. two possible intervals). This gives us the most general kind of rule that we can obtain, and provides a baseline for comparison with frameworks for generating a wider variety of rules.

Definition 8. *Let d be a data item and $(\{\alpha_1, \ldots, \alpha_n\}, \beta)$ be an influence tuple. The* **2-way generalization step** *is the following where for each i, if $v_i > 0.5$, then $\#_i$ is ">", else if $v_i \leq 0.5$, then $\#_i$ is "≤".*

$$\frac{d(\alpha_1) = v_1, \ldots, d(\alpha_n) = v_n, d(\beta) = v_{n+1}}{p(\alpha_1)\#_1 0.5 \wedge \ldots \wedge p(\alpha_n)\#_n 0.5 \rightarrow p(\beta)\#_{n+1} 0.5}$$

For a data item d that satisfies the precondition, then $\mathsf{TwoWayGen}(d, I, \Pi)$ *returns the rule given in the postcondition, otherwise it returns nothing.*

Example 7. *The following is the result of applying the 2-way generalization rule to the data in row 908 in Table 1.*

$$p(\texttt{Qu1}) \leq 0.5 \wedge p(\texttt{Qu3}) > 0.5 \wedge p(\texttt{Enj1}) \leq 0.5 \rightarrow p(\texttt{Pu3}) \leq 0.5$$

Definition 9. *Let D be a dataset, I be an influence tuple, and Π be a set of values. The* **generalize** *function, denoted* $\mathsf{Generalize}(D, I, \Pi)$*, returns the set* $\{\mathsf{TwoWayGen}(d, I, \Pi) \mid d \in D\}$.

Whilst we have focused on a 2-way generalization step, which results in a specific kind of rule, there are various ways we could expand the variety of rules that we could generate from the data. For instance, from the data item d where $d(\texttt{A}) = 0.2$, $d(\texttt{B}) = 0.6$, and $d(\texttt{C}) = 0.9$, we might want to obtain the generalization $p(\texttt{A}) \leq 0.2 \wedge p(\texttt{B}) \geq 0.6 \rightarrow p(\texttt{C}) \geq 0.9$ which involves representation of tighter intervals on belief (less than or equal to 0.2 instead of less than or equal to 0.5, and greater than or equal to 0.6 or 0.9 instead of greater than 0.5).

3.4. Identify the Best and Simplest rules

So from a dataset, an influence tuple, and a value set, we obtain a set of rules. At this stage, these are just candidates, and there is no guarantee that they are good with respect to the data.

Definition 10. *Let Π be a value set. For an atom of the form $p(\beta)\#v$, let* $\mathsf{Values}(p(\beta)\#v, \Pi) = \{x \in \Pi \mid x \# v\}$,

Example 8. *For the rule in Example 7,* $\mathsf{Values}(p(\texttt{Pu3}) \leq 0.5, \Pi) = \{0, 0.5\}$*, where* $\Pi = \{0, 0.5, 1\}$.

In order to harness measures from association rule learning, which we present in Table 2, we require the following subsidiary definitions below. Informally, a rule is fired by a data item when the conditions of the rule are satisfied by the data item. Furthermore, a rule agrees with a data item when the consequent is satisfied by the data item. Finally, a rule is correct with respect to a data item when the rule being fired implies the consequent is satisfied by the data item.

Definition 11. *Let $d \in D$ be a data item, and let $R = \phi_1 \wedge \ldots \wedge \phi_n \rightarrow \phi_{n+1}$ be a rule, and for $i \in \{1, \ldots, n+1\}$, let ϕ_i be of the form $P(\alpha_i)\#_i v_i$. We say R is* **fired** *by d iff for each ϕ_i s.t. $i \leq n$, $d(\alpha_i) \in \mathsf{Values}(P(\alpha_i)\#_i v_i, \Pi)$; R* **agrees** *with d iff $d(\alpha_{i+1}) \in \mathsf{Values}(\phi_{n+1}, \Pi)$; and R is* **correct** *w.r.t. d iff if R is fired by d, then R agrees with d.*

Measure	Definition
$\mathsf{Support}(R, D)$	$\frac{1}{\lvert D\rvert} \times \lvert\{d \in D \mid R \text{ is fired by } d\}\rvert$
$\mathsf{Confidence}(R, D)$	$\frac{1}{\lvert D\rvert} \times \lvert\{d \in D \mid R \text{ is correct w.r.t. } d\}\rvert$
$\mathsf{Lift}(R, D)$	$\frac{\lvert\{d \in D \mid R \text{ is correct w.r.t. } d\}\rvert}{\lvert\{d \in D \mid R \text{ is fired by } d\}\rvert \times \lvert\{d \in D \mid R \text{ agrees with } d\}\rvert}$

Table 2. Measures for support, accuracy and lift where R is a rule, and D is a dataset.

$\mathsf{Generate}(D, I, \tau_{\mathsf{support}}, \tau_{\mathsf{accuracy}})$
$\quad AllRules = \mathsf{Generalize}(D, I, \Pi)$
$\quad BestRules = \mathsf{Best}(AllRules, D, \tau_{\mathsf{support}}, \tau_{\mathsf{accuracy}})$
$\quad return\ \mathsf{Simplest}(BestRules)$

Figure 3. The **generate algorithm** where D is a dataset, I is a set of influence tuples, Π is a value set, $\tau_{\mathsf{support}} \in [0, 1]$ (resp. $\tau_{\mathsf{confidence}} \in [0, 1]$) is a threshold for support (resp. confidence).

Example 9. *Consider the rule* $P(\mathtt{Pu3}) \leq 0.5 \Rightarrow P(\mathtt{Enj1}) > 0.5$ *with data from Table 1. The rule is fired with 903, 908, and 909, and is correct with 903 and 909.*

Given a set of rules and a dataset, the best rules are those that exceed the thresholds for support and confidence and have lift greater than 1.

Definition 12. *For a set of rules Rules, and a dataset D, with a threshold for support* $\tau_{\mathsf{support}} \in [0, 1]$*, and a threshold for confidence* $\tau_{\mathsf{confidence}} \in [0, 1]$*, the set of* **best rules***, denoted* $\mathsf{Best}(Rules, D, \tau_{\mathsf{support}}, \tau_{\mathsf{confidence}})$*, is* $\{R \in Rules \mid \mathsf{Support}(R, D) > \tau_{\mathsf{support}}$ *and* $\mathsf{Confidence}(R, D) > \tau_{\mathsf{confidence}}$ *and* $\mathsf{Lift}(R, D) > 1\}$.

For a set of rules with a particular head, the simplest are those with a minimal set (w.r.t. set inclusion) of conditions. The following Simplest function collects the simplest rules for each head in a set of rules.

Definition 13. *For a rule* $R = \phi_1 \wedge \ldots \wedge \phi_n \rightarrow \psi$*, let* $\mathsf{Conditions}(R) = \{\phi_1, \ldots, \phi_n\}$ *and* $\mathsf{Head}(R) = \psi$*. For a set of rules Rules, the* **simplest rules***, denoted* $\mathsf{Simplest}(Rules)$*, is the set of rules* $\{R \in Rules \mid$ *for all* $R' \in Rules$*, if* $\mathsf{Head}(R) = \mathsf{Head}(R')$*, then* $\mathsf{Conditions}(R) \subseteq \mathsf{Conditions}(R')\}$.

The algorithm for generating the rules is given in Figure 3, and we evaluate a Python implementation[1] of the algorithm in the next section.

4. Evaluation

In this paper, we consider data from two published studies. The data from each study contains the answers from asking individuals a number of questions including their level of agreement with certain statements (as illustrated in Table 1).

[1]Code available at http://www0.cs.ucl.ac.uk/staff/A.Hunter/papers/epilearn.zip

So each row in the data concerns an individual. Each statement can be regarded as an argument. The studies are: (1) the appropriateness of Wikipedia in a Spanish higher education institute [16] which was obtained from 901 individuals and involved 26 statements; and (2) views on political issues in Italy [17] which was obtained from 774 individuals and involved 75 statements.

Example 10. *Some of the statement from the Spanish dataset, are* `Pu3` = *"Wikipedia is useful for teaching",* `Qu1` = *"Articles in Wikipedia are reliable",* `Qu3` = *"Articles in Wikipedia are comprehensive",* `Enj1` = *"Articles in Wikipedia stimulate curiosity",* `Use2` = *"I use Wikipedia as a platform to develop educational activities with students",* `Use3` = *"I recommend my students to use Wikipedia",* `Bi1` = *"In the future, I will recommend the use of Wikipedia to my colleagues and students", and* `Bi2` = *"In the future, I will use Wikipedia in my teaching activities".*

Example 11. *Some of the statements from the Italian dataset, are* `Sys2` = *"In general, the political system works as it should",* `Sys3` = *"The Italian society must be radically changed",* `Sys7` = *"Our society gets worse year by year",* `Sys8` = *"Our society is organized so that people generally get what they deserve",* `Dw6` = *"Every day as society become more lawless and bestial, a person's chances of being robbed, assaulted, and even murdered go up and up", and* `Dw8` = *"It seems that every year there are fewer and fewer truly respectable people, and more and more persons with no morals at all who threaten everyone else".*

For each dataset, we constructed a set of influence tuples by hand based on the text descriptions given for each argument (i.e. the statement) considered in the study. Then, for each influence tuple, we used our algorithm to generate the constraints using the training data (which was a randomly selected subset of 80% of the dataset), and to avoid over-fitting, a maximum of 4 conditions per rule. We evaluated the rules for support, confidence, and lift (as defined in Table 2) using the remaining 20% of the data. Some rules learned from the Spanish (respectively Italian) dataset are given in Example 12 (respectively Example 13).

Example 12. *The following are some of the rules generated from the Spanish dataset, with influence tuple* $(\{\texttt{Qu1}, \texttt{Qu3}, \texttt{ENJ1}, \texttt{JR1}, \texttt{JR2}, \texttt{SA1}\}, \texttt{Pu3})$

1. $p(\texttt{Qu3}) > 0.5 \wedge p(\texttt{Qu1}) > 0.5 \rightarrow p(\texttt{Pu3}) > 0.5$
2. $p(\texttt{Enj1}) \leq 0.5 \wedge p(\texttt{Qu1}) \leq 0.5 \rightarrow p(\texttt{Pu3}) \leq 0.5$
3. $p(\texttt{Jr2}) \leq 0.5 \wedge p(\texttt{Enj1}) \leq 0.5 \rightarrow p(\texttt{Pu3}) \leq 0.5$

Example 13. *The following are some of the rules generated from the Italian dataset, with the influence tuples* $(\{\texttt{Sys1}, \texttt{Sys3}, \texttt{Sys4}, \texttt{Sys5}, \texttt{Sys6}, \texttt{Sys7}, \texttt{Sys8}\}, \texttt{Sys2})$ *and* $(\{\texttt{Sys1}, \texttt{Sys2}, \texttt{Sys4}, \texttt{Sys5}, \texttt{Sys6}, \texttt{Sys7}, \texttt{Sys8}\}, \texttt{Sys3})$.

1. $p(\texttt{Sys7}) > 0.5 \rightarrow p(\texttt{Sys2}) \leq 0.5$
2. $p(\texttt{Sys8}) > 0.5 \rightarrow p(\texttt{Sys2}) \leq 0.5$
3. $p(\texttt{Sys7}) > 0.5 \rightarrow p(\texttt{Sys3}) > 0.5$

For each constraint generated by our algorithm, we tested it using the testing data (i.e. the subset of the dataset after subtracting the training data). We ran

Study	Influence target	No. of influencers	No. of of rules	Condi-tions	Support	Confi-dence	Lift	Time (sec)
Spain	`Use2`	19	11.3	1.0	0.68	0.95	1.04	192.34
Spain	`Use3`	19	14.0	1.69	0.60	0.84	1.16	178.36
Spain	`Bi1`	17	15.8	1.84	0.54	0.82	1.15	148.02
Spain	`Bi2`	17	12.7	2.1	0.51	0.80	1.20	140.98
Spain	`Qu1`	13	3.3	2.07	0.51	0.84	1.37	56.55
Spain	`Qu3`	13	4.2	1.68	0.58	0.88	1.17	48.66
Italy	`Dw1`	9	3.1	2.45	0.43	0.80	1.22	14.33
Italy	`Dw3`	9	4.0	1.0	0.75	0.84	1.15	15.39
Italy	`Dw6`	9	5.0	1.02	0.69	0.88	1.11	17.95
Italy	`Dw8`	9	4.2	1.7	0.67	0.83	1.22	16.65
Italy	`Sys2`	7	7.0	1.0	0.76	0.96	1.03	7.89
Italy	`Sys3`	7	1.6	1.48	0.52	0.82	1.22	8.21

Table 3. Results for the Spanish and Italian datasets with 10 repetitions. Column 3 is the number of influencers in the influence tuple. Column 5 is the average number of conditions per rule. For columns 4 to 9, the value is the average of the repetitions with $\tau_{\mathsf{confidence}} = 0.8$ and $\tau_{\mathsf{support}} = 0.4$.

the Python implentation in an evaluation on a Windows 10 HP Pavilion Laptop (with AMD A10 2GHz processor and 8GB RAM). In Table 3, we give results for some influence tuples with the Spanish and Italian datasets.

These results show that we are able to obtain reasonable quality constraints (in terms of support, confidence, and lift) from data by our simple version of association learning. Furthermore, the number of rules selected, and the complexity of those rules, tend to be appropriate for the application (i.e. enough rules to give insights into the data but still reasonably concise). Also, the number of rules and the average measure of support per rule are quite high (for example, for `Use2` in Table 3, it is 11.3 and 0.68 respectively) which means that the data shows that there is indeed a number of ways that the target is influenced by other arguments, and each of those ways occurs frequently. Note, we have set a quite high threshold for support, and by lowering this, we can raise lift above 2.

The time performance is also reasonable. For the Spanish dataset, we consider sets of influencers of cardinality of up to 19 arguments. This means that a large number of rules can be constructed for each subset of influencers (e.g. over 70K for 19 influencers with a maximum of 4 conditions per rule). Yet the number of rules returned by the algorithm is often in the range of 10 to 20 rules, and the algorithm is running in less than 200 seconds. Furthermore, it is reasonable to expect that for many domains we should be able to restrict the number of influencers for each argument to less than 20 (and compare this with most argument graphs where far fewer attackers per argument are represented).

5. Comparison with the Literature

Two important approaches to probabilistic (abstract) argumentation are the constellations and the epistemic approaches [7]. In the constellations approach, there

is uncertainty about which arguments and attacks should appear in the argument graph [5,14]. In contrast, in the epistemic approach, the topology of the argument graph is fixed, but there is uncertainty about whether an argument is believed [20,7,2,6,12]. The approach of epistemic graphs is a generalization of the epistemic approach.

For some time, there has been interest in using argumentation for improving machine learning and using machine learning for generating arguments (for a review, see [3]). In the literature, there are three recent proposals for learning for argumentation that are based on probabilistic techniques, though they are different to our proposal. The first proposal uses the usual labels for arguments *in*, *out* and *undecided*, augmented with *off* for denoting that the argument does not occur in the graph [19]. A probability distribution over labellings gives a form of probabilistic argumentation. For learning, the probability distribution is used to generate labellings that are used as data, and then the argument graph that best describes this data is identified. The second proposal takes as input a profile $\langle X_1, \ldots, X_n \rangle$ where each X_i is a set of acceptable arguments, and by using Bayes theorem, the output is a posterior probability for a set of arguments being an extension. This is calculated using a Bayesian network that incorporates assumptions about the relationships between choice of semantics and choice of attacks, and how these influence extensions [13]. The third proposal generates the probability distribution over subgraphs as used in the constellations approach [9]. It takes as input a profile $[(\phi_1, v_1), \ldots, (\phi_n, v_n)]$ where each ϕ_i is a Boolean combination of arguments that specifies an opinion on the topology of the argument graph, and v_i is the belief in that opinion, and returns the probability distribution that best represents the opinions. These three proposals concern uncertainty about the structure of the graph. Clearly none involve the epistemic approach to probabilistic argumentation, and in particular, none consider how constraints for epistemic graphs could be obtained from data.

6. Discussion

In this paper, we have proposed a framework for learning a class of constraints for epistemic graphs and evaluated it with two datasets. Generating epistemic graphs for argumentation offers a valuable way of constructing a representation of how arguments interact. A significant barrier to the deployment of argumentation formalisms has been the challenge of how to construct the required representations. Taking a probabilistic approach allows us to overcome this hurdle and thereby scale up the kind of problem we can tackle with an argumentation solution.

From the point of view of association rule learning [1], we have only presented a very simple framework to show that it is viable to generate constraints for epistemic graphs in this way. In future work, we will introduce alternatives to the 2-way generalization step so that we can learn a wider variety of rules from the restricted epistemic language presented using tight constraints and a wider variety of values as discussed at the end of Section 3.3, and more complex rules such as with heads that provide both upper and lower bounds on belief (e.g. $p(\texttt{Sys7}) > 0.5 \rightarrow p(\texttt{Sys3}) > 0.5 \wedge p(\texttt{Sys3}) \leq 0.7$). We will also consider a less

restricted version of the language of epistemic graphs (i.e. use the full language as defined in [11]), and we will consider how we can learn labels for the epistemic graphs.

References

[1] R. Agrawal, T. Imieliski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of SIGMOD '93*, pages 207–216. ACM Press, 1993.

[2] P. Baroni, M. Giacomin, and P. Vicig. On rationality conditions for epistemic probabilities in abstract argumentation. In *Proceedings of COMMA'14*, 2014.

[3] O. Cocarascu and F. Toni. Argumentation for machine learning: A survey. In *Proceedings of COMMA'16*, pages 219–230. IOS Press, 2016.

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

[5] P. M. Dung and P. M. Thang. Towards (probabilistic) argumentation for jury-based dispute resolution. In *Proceedings of COMMA'10*, pages 171–182, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[6] D. Gabbay and O. Rodrigues. Probabilistic argumentation: an equational approach. *Logica Universalis*, 9(3):345–382, 2015.

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

[8] A. Hunter, L. Chalaguine, T. Czernuszenko, E. Hadoux, and S. Polberg. Towards computational persuasion via natural language argumentation dialogues. In *Proceedings of KI'19*, volume 11793 of *LNCS*, pages 18–33. Springer, 2019.

[9] A. Hunter and K. Noor. Aggregation of perspectives using the constellations approach to probabilistic argumentation. In *Proceedings of AAAI'20*, pages 2846–2853. AAAI Press, 2020.

[10] A. Hunter, S. Polberg, and N. Potyka. Updating Belief in Arguments in Epistemic Graphs. In *Proceedings of KR'18*, pages 138–147. AAAI Press, 2018.

[11] A. Hunter, S. Polberg, and M. Thimm. Epistemic graphs for representing and reasoning with positive and negative influences of arguments. *Artificial Intelligence*, 281:103236, 2020.

[12] A Hunter and M Thimm. Probabilistic reasoning with abstract argumentation frameworks. *Journal of Artificial Intelligence Research*, 59:565–611, 2017.

[13] H. Kido and K. Okamoto. A Bayesian approach to argument-based reasoning for attack estimation. In *Proceedings of IJCAI'17*, pages 249–255. IJCAI, 2017.

[14] H. Li, N. Oren, and T. Norman. Probabilistic argumentation frameworks. In *Proceedings of TAFA'11*, volume 7132 of *LNCS*, pages 1–16. Springer, 2012.

[15] R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 140:1–55, 1931.

[16] A. Meseguer-Artola, E. Aibar, J. Llads, J. Minguilln, and M. Lerga. Factors that influence the teaching use of Wikipedia in higher education. *Journal of the Association for Information Science and Technology*, 67(5):1224–1232, 2016.

[17] V. Pellegrini, L. Leone, and M. Giacomantonio. Dataset about populist attitudes, social world views, socio-political dispositions, conspiracy beliefs, and anti-immigration attitudes in an italian sample. *Data in Brief*, 25:104144, 2019.

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

[19] R. Riveret and G. Governatori. On learning attacks in probabilistic abstract argumentation. In *Proceedings of AAMAS'16*, pages 653–661, 2016.

[20] M. Thimm. A probabilistic semantics for abstract argumentation. In *Proceedings of ECAI'12*. IOS Press, 2012.