Using MetaProbLog and ConArg to compute Probabilistic Argumentation Frameworks

Stefano Bistarelli[1], Theofrastos Mantadelis[1], Francesco Santini[1], and Carlo Taticchi[2]

[1] Department of Mathematics and Computer Science
University of Perugia, Italy
{firstname.lastname}@unipg.it
[2] Gran Sasso Science Institute
L'Aquila, Italy
carlo.taticchi@gssi.it

Abstract. In Probabilistic Abstract Argumentation, arguments and attacks (nodes and edges) in a graph instance are associated with a probability value. These probabilities can be interpreted in different ways: for instance, in the constellation approaches, the probabilities introduce uncertainty in the topology of the graph. In this paper we use MetaProbLog, a ProbLog framework where facts in a logic program are annotated by probabilities; the purpose is to compute the probability of possible worlds of arguments. The tool is integrated in the web interface of ConArg, a constraint-programming based tool aimed to solve different problems in Abstract Argumentation.

1 INTRODUCTION

Argumentation is a method of humanity to discuss and solve myriad different situations where points of views may conflict. In AI, *Abstract Argumentation* [4] aims to provide a model abstracting from the underlying logic and inference process. In this field, defining *semantics* amounts to specifying a declarative or procedural method to derive a set of argument subsets (i.e., *extensions*) from an *Abstract Argumentation Framework* (*AAF*) [4], which is simply defined by a set of arguments and an attack relationship.

Knowledge representation with the use of probabilistic information has been used in many areas of Computer Science, and it is a powerful medium to represent knowledge. For this reason, many researchers have extended AAFs by adding probabilistic information. These very prominent extensions of AAFs have been categorized in two big groups by Hunter [7]: the **epistemic** and the **constellation** approaches.

The epistemic approaches, such as those presented in [16] describe probabilistic AAFs where the uncertainty does not alter the structure of the AAFs. This type of AAFs use the probability assignments to quantify the existing uncertainty of arguments in AAFs and not to introduce new uncertainty.

The constellation approaches, such as those presented in [10] introduce probabilistic values that are associated with the elements in an AAF; in such a way, the uncertainty related to the structure of the AAF can be represented. The constellation approaches generate a set of AAFs with a probabilistic distribution and as such they define a probabilistic distribution over the extensions of those AAFs. Fazzinga et al. [5] discuss the complexity for computing different semantics in PrAAFs.

2 BACKGROUND

An abstract argumentation framework [4] (AAF) is a tuple $AAF = (Args, Atts)$ where $Args$ is a set of arguments and $Atts$ a set of attacks among arguments of the form of a binary relation $Atts \subseteq Args \times Args$. For $a, b \in Args$, we use $a \rightarrow b$ as a shorthand to indicate $(a, b) \in Atts$ and we say that argument a *attacks* argument b.

A set of arguments $S \subseteq Args$ is said to be *conflict-free* iff $\nexists a, b \in S$ where $a \rightarrow b$. An argument $a \in Args$ is acceptable with respect to set $S \subseteq Args$ if no argument attacks a or if $\forall b \in Args$ that $b \rightarrow a$ then $\exists c \in S$ where $c \rightarrow b$.

Dung [4] further gives semantics to AAF by the use of extensions over subsets of arguments. Dung first defines the *admissible* semantics. A set $S \subseteq Args$ is admissible iff S is conflict free and each $a \in S$ is acceptable with respect to S. Over time several semantics have been discussed such as complete, preferred, grounded, stable, etc.

Hunter [7], categorizes probabilistic abstract argumentation frameworks (PrAAFs) in two different categories: the *constellation* and the *epistemic* PrAAFs. For this paper we will focus on the constellation approaches and we base our work in the definition of PrAAFs by [10]. A constellation approach to PrAAFs defines probabilities over the structure of the AAF graph. One can assign probabilities to either the arguments or/and attacks of the AAF. We refer to arguments/attacks with assigned probabilities less than 1 as probabilistic arguments/attacks. For this work we restrict PrAAFs to have probabilities attached only in attacks. In [11] is shown that PrAAFs with probabilities only to attacks can represent any general PrAAF as defined by [10]. A probabilistic attack $a \rightarrow b$ exists in an AAF with probability $P(a \rightarrow b)$. These probabilistic attacks correspond to random variables, which are assumed to be mutually independent. As such, a PrAAF defines a probability distribution over a set of AAFs.

Definition 1. *Formally, a PrAAF is a tuple $PrAAF = (Args, Atts, P_{Atts})$ where $Args$, $Atts$ define an AAF, P_{Atts} is a set of probabilities for each $\rightarrow \in Atts$ with* $0 < P_{Atts}(\rightarrow) \leq 1$.

Finally, stating an attack has probability 0 is redundant. A probabilistic attack with 0 probability is not part of any AAF that the constellation represents and is omitted.

A PrAAF defines a probability distribution for all the possible non-probabilistic AAFs it contains. Each single possible set of probabilistic attacks of the PrAAF is called a **possible world**. The possible worlds of a PrAAF are exponential in the number of probabilistic attacks (2^N where N the number of probabilistic attacks).

Definition 2 (Probability of Possible World). *The probability of a possible world equals to the product of the probability of each probabilistic attack that is in the possible world with the product of one minus the probability of each probabilistic attack that is excluded from the possible world.*

$$P_{world} = \prod_{e_i \in AAF_{world}} P(e_i) \cdot \prod_{e_j \notin AAF_{world}} (1 - P(e_j))$$

The usual AAF semantics are slightly modified in PrAAFs. For example, in PrAAFs the inquisitor is not asking if a set Q is admissible in PrAAF P; but what is the probability that set Q is admissible in PrAAF P, meaning with what probability exists an

AAF where Q is admissible. Similarly, for different semantics than admissible such as complete, preferable, etc.

In this work [1] we only focus on enumerate inference. Our system computes the probability of a user given set Q^3 being within the admissible or conflict free semantics under the enumerate inference. We leave for future work the computation of arguments being credulously or skeptically admissible or conflict free.

3 Implementation

We use *MetaProbLog* [13][4], a framework based on ProbLog [8] probabilistic logic programming language. ProbLog extends Prolog programs by annotating facts with probabilities. In that way it defines a probability distribution over all Prolog programs. ProbLog follows the distribution semantics presented by Sato [15]. MetaProbLog extents ProbLog with high order calls. We use MetaProbLog to model the constellation approach, and we integrate its use through the web interface of ConArg[5] [3]. As far as we know, what we present in this paper is the first application of this kind which is openly available to the scientific community. Other attempts which are not available online include [10, 6].

MetaProbLog provides several different efficient probabilistic inference methods such as: (i) exact inference based on Reduced Ordered Binary Decision Diagrams (ROBDDs) and dynamic programming [12, 8]; (ii) program (AAF) sampling with memoization [9]; (iii) any-time inference using an iterative deepening algorithm [14].

The web interface exposes two forms of probabilistic inference: exact and AAF sampling. The exact inference computes the exact probability; the AAF sampling inference is an approximation method. In most cases the exact inference is able to compute the result faster than most approximation methods, such as the AAF sampling inference. But exist cases where exact inference is intractable and a user is forced to use an approximation method, for those cases we provide the AAF sampling inference.

The program sampling inference is based on the use of Monte Carlo methods, that is, to use the ProbLog program to generate large numbers of random subprograms[6] and to use those to estimate the probability. The implementation of this approach for MetaProbLog, is similar with the one described at [9], and takes advantage of the independence of probabilistic facts to generate samples lazily while proving the query, that is, sampling and searching for proofs which are interleaved.

We integrated MetaProbLog with the web interface[7] of ConArg (Figure 1) in order to directly take input from the interface and return the probability of queries of PrAAFs. The user can select “probabilistic” from the ConArg web interface panel in order to start working with PrAAFs, three main tasks can be performed with the aid of the tool: first of all, a framework whose attacks are endowed with a probabilistic value can be given

[3] There exist an exponential number of sets Q to the number of arguments that have $0 < P_{sem}(a \in Q) < 1$. For that reason we do not enumerate all.
[4] MetaProbLog is available at: `www.dcc.fc.up.pt/metaproblog`
[5] `http://www.dmi.unipg.it/conarg/`.
[6] We note that, each sampled ProbLog program corresponds to sampling an AAF.
[7] `http://www.dmi.unipg.it/conarg/`

as input and visualized directly in the interface; then the left menu can be used to query for extensions w.r.t. a given semantics; finally the results of the query are shown both as text and visually on the represented graph. We detail each of these tasks separately.

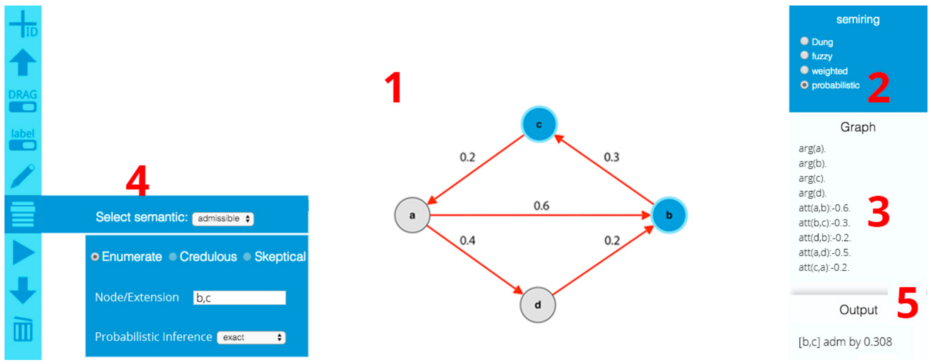

Fig. 1. The ConArg Web Interface. The marked areas correspond to: 1) Example of a PrAAF in the ConArg web interface. Nodes highlighted in blue represent the subset selected to be checked w.r.t. a semantics. 2) Semiring selection panel. 3) The specification of the PrAAF, with probabilities attached to attacks. 4) The query panel to select the semantics and specify a node or an extension to test, along with the type of probabilistic inference. 5. Outputs the result of the query.

Representation. A visual representation is helpful to study and understand properties of AAFs or to look for counterexamples. The input PrAAF is entered by drawing on the interface or by typing it in the text box of the right panel (Figure 1.3), with the following syntax: `arg(a)` defines an argument a, and `att(a,b):-0.6` denotes that an attack from a to b exists with probability 0.6. The visual and the textual representations are consistent with each other so that a change in one will modify the other accordingly.

Queries. Due to the probabilistic nature of PrAAFs, looking for an extension w.r.t. a semantics means investigating the probability that a set of arguments belongs to that semantics. Once the tool has received the PrAAF in input, it is possible to test a subset of arguments in order to obtain the probability with which it is in a certain semantics. All the settings for the query can be configured in a panel of the left menu (Figure 1.4). The “Select semantics” options list allows to choose between the conflict free and the admissible semantics, while the arguments can be listed in the “Node/Extension” text field. Moreover, we can specify which type of probabilistic inference has to be used for computing the solution: either “exact” or with “AAF sampling”.

Output. The output can be read on the right panel of the web interface (Figure 1.5). For instance, the string `Set [b,c] is admissible by 0.308` is the output for the query given in the paragraph above as an example, and expresses the fact that the subset $\{b, c\}$ of the PrAAF is admissible with a probability of 0.308.

4 Conclusion

In this paper we presented the integration of constellation PrAAFs into the web interface of ConArg [3]. For the probabilistic inference we used MetaProbLog [13, 14]. In the constellation approaches, uncertainty in the topology of the graph (probabilities on arguments and attacks) is used to make probabilistic assessments on the acceptance of arguments. As far as we know, this is the first tool based on the constellation approach that is openly offered to the scientific community through a web interface. The goal of this work is to further enrich ConArg and to provide to the scientific community a wider set of problems that can be solved. As future work we plan to also extent the ConArg library [2] to support probabilistic inference.

Acknowledgements

This work has been supported by: "ComPAArg" (Ricerca di base 2016–2018), "Argumentation 360" (Ricerca di Base 2017–2019) and "RACRA" (Ricerca di base 2018–2020).

References

1. Bistarelli, S., Mantadelis, T., Santini, F., Taticchi, C.: Probabilistic Argumentation Frameworks with MetaProbLog and ConArg. In: International Conference on Tools with Artificial Intelligence, (ICTAI). pp. 675–679 (2018)
2. Bistarelli, S., Rossi, F., Santini, F.: A conarg-based library for abstract argumentation. In: International Conference on Tools with Artificial Intelligence, (ICTAI). pp. 374–381 (2017)
3. Bistarelli, S., Santini, F.: Modeling and solving afs with a constraint-based tool: Conarg. In: Theorie and Applications of Formal Argumentation, (TAFA). pp. 99–116 (2011)
4. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intel. **77**(2), 321–358 (1995)
5. Fazzinga, B., Flesca, S., Parisi, F.: On the complexity of probabilistic abstract argumentation. In: International Joint Conference on Artificial Intelligence. pp. 898–904 (2013)
6. duy Hung, N.: Probabilistic argumentation for decision making: A toolbox and applications. Journal of Intelligent Informatics and Smart Technology (2016)
7. Hunter, A.: Some foundations for probabilistic abstract argumentation. In: Computational Models of Argument, (COMMA). pp. 117–128 (2012)
8. Kimmig, A., Demoen, B., De Raedt, L., Santos Costa, V., Rocha, R.: On the implementation of the probabilistic logic programming language ProbLog. Theory and Practice of Logic Programming (TPLP) **11**(2-3), 235–262 (2011)
9. Kimmig, A., Gutmann, B., Santos Costa, V.: Trading memory for answers: Towards tabling problog. In: International Workshop on Statistical Relational Learning, (SRL) (2009)
10. Li, H., Oren, N., Norman, T.J.: Probabilistic argumentation frameworks. In: Theorie and Applications of Formal Argumentation, (TAFA). pp. 1–16 (2011)
11. Mantadelis, T., Bistarelli, S.: A Preliminary Report on Probabilistic Attack Normal Form for Constellation Semantics. ArXiv e-prints (Sep 2018)
12. Mantadelis, T., Janssens, G.: Dedicated tabling for a probabilistic setting. In: International Conference on Logic Programming, ICLP. Leibniz International Proceedings in Informatics, (LIPIcs), vol. 7, pp. 124–133 (2010)
13. Mantadelis, T., Janssens, G.: Nesting probabilistic inference. In: International Colloquium on Implementation of Constraint LOgic Programming Systems (CICLOPS) (2011)
14. Mantadelis, T., Rocha, R.: Using iterative deepening for probabilistic logic inference. In: Practical Aspects of Declarative Languages, (PADL). pp. 198–213 (2017)
15. Sato, T.: A statistical learning method for logic programs with distribution semantics. In: International Conference on Logic Programming, (ICLP). pp. 715–729 (1995)
16. Thimm, M.: A probabilistic semantics for abstract argumentation. In: European Conference on Artificial Intelligence, (ECAI). pp. 750–755 (2012)