Towards Learning Argumentation Frameworks from Labelings

Lars Bengel[0000−0003−0360−0485]

Artificial Intelligence Group, University of Hagen, Germany
lars.bengel@fernuni-hagen.de

1 Introduction

An abstract argumentation framework (AF) due to Dung [1] is defined as a graph $F = (\mathsf{Arg}, \mathsf{R})$, where the nodes are *arguments* and the edges represent *attacks* between these arguments. An attack from an argument A to another argument B means that, if we consider the argument A to be acceptable, then we have to reject B, since A contradicts B. The goal of this approach is to model human argumentation in a formal manner and to use this model for reasoning.

A central notion in abstract argumentation is that of an *(argumentation) semantics*. A semantics σ characterizes sets of arguments (called *extensions*) that are jointly acceptable. In particular, they usually require that extensions are *conflict-free*, i. e., that there is no conflict between arguments of the extension, and that it defends itself, i.e., it counterattacks all its attackers (the latter property is called *admissibility*). Based on these notions, we can define different semantics, like e. g., complete semantics, where all arguments defended by a set E have to be contained in E.

We are especially interested in labeling-based semantics [2]. Instead of just providing sets of acceptable arguments, they determine *labelings* that assign to each argument one of three labels. If an argument is considered acceptable it is labeled in, if the argument is attacked by some acceptable argument it gets the label out and, otherwise, it receives the label undec.

In general, semantics allows us to perform *inference*, by determining semantical information (extensions or labelings) from given syntactical information (in the form of an AF). Using, e. g., the labelings of an AF we can then draw conclusions from the framework by considering the accepted arguments. We are however interested in the reverse direction, i. e., the process of *inductively learning* a syntactic structure from semantical information. This process can be considered as a form of inductive reasoning, where we generalize from observations (in the form of a given set of labelings) to a suitable AF that explains this input. This AF then also enables further reasoning possibilities, e. g., computing additional compatible labelings.

Assume a scenario, where we can discuss with a person about their beliefs on a specific subject. From this discussion, we can obtain knowledge about their beliefs, in the form of labelings. Now, to gain a better understanding of their internal reasoning, we want to learn AFs that are compatible with these labelings.

Knowing the graph structure that is consistent with a labeling is helpful in making the labeling explainable. This may also allow us to construct better counterarguments in order to persuade them to change their beliefs.

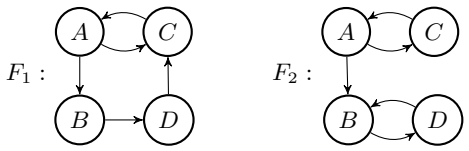

Fig. 1. Some AFs that are compatible with the complete labeling ℓ.

Example 1. Consider the complete labeling $\ell = \{\mathsf{in} : \{A, D\}, \mathsf{out} : \{B, C\}, \mathsf{undec} : \emptyset\}$. Both AFs in Figure 1 are compatible and can be used to explain ℓ. For example, F_1 would tell us that D may be accepted despite the attack $B \to D$, because A defends D against B. That also means, constructing an argument that refutes A would be very effective in challenging a person that beliefs in ℓ. This information is not immediately apparent from the labeling ℓ alone.

2 Approach

There exist few approaches that address the problem of learning AFs from labelings or similar problems [3, 4]. However, neither of them fully addresses the scenario outlined above. The main issue with these approaches is, they only compute a single solution for an input, while in reality there might be multiple compatible AFs, e. g., the AFs in Figure 1 are both compatible with ℓ. Therefore, we propose a new algorithm that, given a set of labelings and associated semantics, computes so-called *attack constraints* for each argument. For example, given an argument in a complete input labeling ℓ, the constraints are computed according to the following method.

$$AttCon_{co}(a, \ell) = \begin{cases} \bigwedge\limits_{b \in \mathsf{Arg} \setminus \mathsf{out}(\ell)} \neg r_{ba} & \text{if } a \in \mathsf{in}(\ell) \\ \bigvee\limits_{b \in \mathsf{in}(\ell)} r_{ba} & \text{if } a \in \mathsf{out}(\ell) \\ \bigwedge\limits_{b \in \mathsf{in}(\ell)} \neg r_{ba} \wedge \left(\bigvee\limits_{c \in \mathsf{undec}(\ell)} r_{ca} \right) & \text{if } a \in \mathsf{undec}(\ell) \end{cases} \tag{1}$$

The atoms of these formulas directly correspond to attacks in AFs and thus allow for an efficient representation of the set of all AFs that are compatible with the input. Moreover, with this approach, we can easily incorporate additional labelings and refine our result. Current work includes elaborating this idea and conducting an experimental study of its feasibility.

Acknowledgements The research reported here was partially supported by the Deutsche Forschungsgemeinschaft (grant 375588274).

References

1. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial intelligence **77**(2), 321–357 (1995)
2. Caminada, M.W., Gabbay, D.M.: A logical account of formal argumentation. Studia Logica **93**(2), 109–145 (2009)
3. Riveret, R., Governatori, G.: On learning attacks in probabilistic abstract argumentation. In: Proceedings of the AAMAS 2016. pp. 653–661 (2016)
4. Niskanen, A., Wallner, J., Järvisalo, M.: Synthesizing argumentation frameworks from examples. Journal of Artificial Intelligence Research **66**, 503–554 (2019)