Argumentation-based Explainable Machine Learning ArgEML: α-Version Technical Details

Nicoletta Prentzas[1]

[1] Department of Computer Science, University of Cyprus, 1 University Avenue, 2109 Aglantzia, Cyprus

Abstract

The paper presents the technical details of the ArgEML system α-version, which implements a general argumentation-based framework and methodology for Explainable Machine Learning. ArgEML is based on a novel approach that integrates sub-symbolic methods with logical methods of argumentation to provide explainable solutions to learning problems.

Keywords

Explainable Machine Learning, Argumentation in Machine Learning, Explainable Conflict Resolution.

1. ArgEML Framework

ArgEML is motivated by several works in the literature that explore the potential of the strong connection of argumentation with learning in the context of explainability. Some of these works have studied how to learn argumentation frameworks from data, abstract frameworks, [1], [2],[3],[4],[5] or structured frameworks, [6], [7], [8], [9], [10], [11]. Other interesting works can be found in [12], [13], [14], [15],[16], [17], [18], [19].

The ArgEML learning methodology is a case of symbolic supervised learning, that can be also applied in a hybrid mode on top of other symbolic or non-symbolic learners that would generate an initial learning theory. The methodology is outlined in Figure 1 and briefly explained in paragraph 1.1.

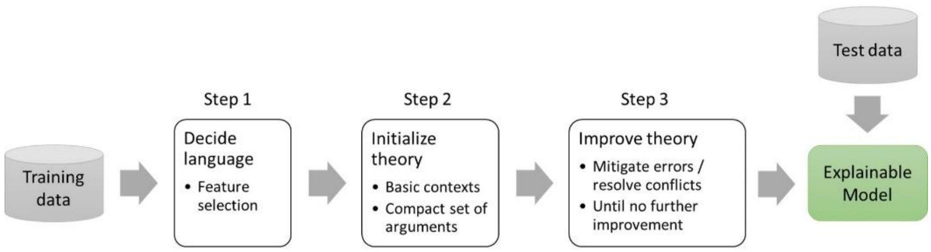

Figure 1: ArgEML Methodology

1.1. Methodology overview

- Step 1: decides the language (relevant features / predictors) of the learning problem in a similar way to the data processing step in a standard machine learning pipeline.
- Step 2: identifies the basic contexts of the problem domain by selecting a compact set of arguments with high coverage to initialize the theory.

Both steps (1) and (2) can be executed automatically or in a hybrid mode by calling onto a sub-symbolic or symbolic existing learner.

Late-breaking work, Demos and Doctoral Consortium, colocated with The 1st World Conference on eXplainable Artificial Intelligence: July 26–28, 2023, Lisbon, Portugal
prentzas.nicoletta@ucy.ac.cy (N. Prentzas)

- Step 3: involves a repeated learning process to produce an argumentation theory as the final output of the learner. At each iteration step two main operators are considered: a mitigation of errors in the definite prediction of (some part of) the current theory and an operator for resolving conflicts in the ambiguity of the current theory. The step is guided by a learning assessment (metric) that measures the quality of a theory as a trade-off between accuracy and ambiguity.

The resulting explainable model is an argumentation theory that supports the conclusions (labels) of a target variable (classification problem case). To generate a prediction for an input case the theory is queried against all possible conclusions. If exactly one conclusion can be derived then the prediction is considered *definite*, otherwise, the conclusion forms a *dilemma* within the theory. Moreover, a definite prediction can be correct or wrong, that is *definite correct* or *definite wrong*. The *learning assessment* metric, which is a generalization of the standard classification accuracy, is defined as:

$$Learning\ Assessment\ (LA)\ =\ \frac{(definite\ correct\ predictions)\ +\ dilemmas * wd}{total\ number\ of\ predictions\ made} \tag{1}$$

LA includes a weighted element w_d that reflects the weakness of dilemmas of the theory, e.g. for binary classification learning problem this factor can be chose to be one-half.

2. ArgEML system: α-version

System components and its main functions are discussed in chapters 2.1 and 2.2 respectively, whereas in chapter 2.3 we explain the evaluation (system verification) process followed. Details of the ArgEML theory and learning method can be found in [20]. Figure 2 shows two screenshots of the system, an ArgEML run on the left, and an ArgEML output on the right.

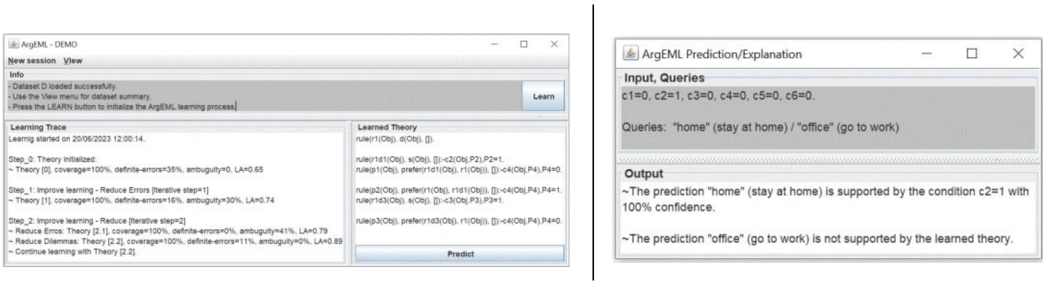

Figure 2: ArgEML system screenshots, an ArgEML run (left), and ArgEML output (right).

2.1. System components

The ArgEML system is a Java application that integrates with Gorgias [21], a structured argumentation framework, for the development and evaluation of the argumentation theory it generates. In the *automatic* mode of operation, the application accepts as input a dataset (examples + feature set), while in the *hybrid* mode of operation, the system also accepts as input the results of an external ML model's execution on the input dataset. The current implementation can process the results of the inTrees [22] library. The application interacts with the SWI-Prolog[2] component for the evaluation of the Gorgias argumentation theories learned. This interaction is achieved via the JPL API[3].

---

[2] A versatile implementation of the Prolog language. https://www.swi-prolog.org/
[3] A library that provides a bidirectional interface between Java and Prolog. https://jpl7.org/ .

2.1. Main functions

The system accepts as input a dataset in the form of a csv file (feature set is automatically derived from the file), a set of decision-rules in a predefined format as a csv file, and a set of parameters that control the learning process. In the *automatic* mode of operation, the learning process starts from exploring the input feature set, to construct the initial set of arguments. In the *hybrid* mode, additional knowledge is provided as input to the system, in the form of association rules between input features and the target feature.

The output of the system is a Gorgias argumentation theory that we can use like any other ML model to generate predictions for new inputs with the corresponding explanations. The execution of the system is highly parametric allowing the end user to fine tune the execution of the process. The basic parameters are shown in Table 1.

Table 1
System parameters

Parameter	Values	Description
Initialize-theory	args-np / args-wp /mixed	Strategy to define type of initial arguments as general or with premises or both.
Definite-errors-threshold	percentage	The target value for Definite errors metric.
Ambiguity-threshold	percentage	The target value for Ambiguity metric.
Majority-class-threshold	percentage	It defines the percentage above which a class of data is considered as a majority class.
Balanced-distribution	percentage	It defines the range up to which a class distribution is considered balanced.
Iterative learning-steps	integer	It defines the maximum number of iterative learning steps.
Data-split (train / test)	percentages	It defines the percentages for splitting the data into train and test.
Rules-complexity	integer	It defines the maximum number of conditions for rules selected by the hybrid process of step 2.
Learning-assessment-loss	decimal<1	It defines the acceptable performance loss during the iterative learning process.

args-np:arguments without premises. args-wp:arguments with premises.

- Parameters fine tuning: The user can experiment with various parameter values to understand under which configuration the system performs better for their problem.
- Explanations (system output): Explanations of a prediction are provided in a natural form containing also a contrastive element against other possible predictions. An example of explanation is shown in Table 2. In this example the system learns an argumentation theory from an artificial dataset with 10 binary features that supports scenarios for “staying at home” or “going to work”.

Table 2
ArgEML example of input / output

Input: {c1=0, c2=1, c3=0, c4=0, c5=0, c6=1, c7=0, c8=0, c9=0, c10=0, target=work}
Output: Prediction: *work*, Explanation: The prediction *work* is supported by the fact c7=0. While the contrary prediction of *home* is also supported by the fact c8=0, the reason of c7=0 supporting *work* is stronger when c3=0. Moreover, although the fact c1=0 could render the argument for *home* based on c8=0 stronger this is not so, because c4=0 holds.

The system can also use the argumentation-based explanations to partition the problem-space into different sub-groups, examples are show in Table 3.

Table 3
ACSRS Example, Explanation Sub-groups

Explanation group	E1	E2	E3	E4.1,E4.2	E5.1,E5.2
Number of cases	44	18	9	5	6
Accuracy / Dilemma	96%	95%	100%	Dilemma	Dilemma

The system can use these sub-groups to provide a grading a confidence for new predictions depending on the group that a new case may fall. Also, the identification of the dilemma groups can guide us to look for new data (to help resolve these).

2.2. Evaluation of the α-version System

Currently, the ArgEML system supports classification problems on datasets with categorical features. The ArgEML system is under continuous evaluation on different learning problems through which we get feedback that can help us tune and improve the approach. We present the results of our experimentation on three datasets, (1) an artificial dataset, (2) a standard dataset from a ML repository, and (3) a real-life image dataset. We compare the results with Random Forest (RF) models in Table 4.

Table 4
ACSRS RF comparison (in terms of accuracy)

Dataset (size)	Parameters [a]	Train set(80%)			Test set(20%)		
		RF	ArgEML		RF	ArgEML	
		CA	DA	LA	CA	DA	LA
Artificial Dataset (120)	{args-wp, 0%, 0%, n/a}	1	1	1	n/a	n/a	n/a
IRIS (150)	{args-np, 0%, 0%, n/a}	0.96	0.96	0.96	0.90	0.93	0.93
ACSRS (200)	{hybrid, 5%, 10%, 2}	0.90	0.94	0.84	0.78	0.77	0.71

a:{initialize-theory, definite-errors-threshold, ambiguity-threshold, rules-complexity}.
All experiments run with majority-class=60%, balanced-distribution=20%, iterative-learning-steps=10. The experiment on the artificial dataset run with 100% on the train set. RF: Random Forest. CA: Classification Accuracy. LA: Learning Assessment. A: Definite Accuracy.

The comparison shown in Table 4 is between the metric of *Definite Accuracy*, defined as (definite correct predictions) / (definite predictions), for the ArgEML theories, and *Classification Accuracy* for the RF models. We are also currently experimenting by running ArgEML in hybrid mode on top of standard explainability systems, such as LIME [23], SHAP [24] and GLocalX [25].

3. Contribution to xAI community

The related material and the codebase of the system, together with the example data sets used in the demo are available on GitHub (github.com/nicolepr/argeml). The release of ArgEML α-version will provide the research community with another xAI tool for learning, experimentation and development of explainable solutions for decision support. We look forward to collaborate with the community to improve ArgEML and also work on new ideas. An important case of this is to examine how ArgEML can be used to enhance post-hoc explainability layer for opaque black-box learned models.

References

[1] A. Niskanen, J. P. Wallner, and M. Järvisalo, "Synthesizing argumentation frameworks from examples," *J. Artif. Intell. Res.*, vol. 66, 2019, doi: 10.1613/jair.1.11758.

[2] K. Č Yras, K. Satoh, and F. Toni, "Abstract argumentation for case-based reasoning," *Proc. Int. Conf. Knowl. Represent. Reason.*, no. Kr, pp. 549–552, 2016.

[3] H. Ayoobi, M. Cao, R. Verbrugge, and B. Verheij, "Argumentation-Based Online Incremental Learning," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, 2022, doi: 10.1109/TASE.2021.3120837.

[4] N. Potyka, M. Bazo, J. Spieler, and S. Staab, "Learning Gradual Argumentation Frameworks using Meta-heuristics," in *CEUR Workshop Proceedings*, 2022, vol. 3208.

[5] N. Potyka, "Interpreting Neural Networks as Quantitative Argumentation Frameworks," in *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, 2021, vol. 7, doi: 10.1609/aaai.v35i7.16801.

[6] Y. Dimopoulos and A. Kakas, "Learning non-monotonic logic programs: Learning exceptions," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1995, vol. 912, pp. 122–137, doi: 10.1007/3-540-59286-5_53.

[7] M. Wardeh, F. Coenen, and T. B. Capon, "PISA: A framework for multiagent classification using argumentation," *Data Knowl. Eng.*, vol. 75, pp. 34–57, 2012, doi: 10.1016/j.datak.2012.03.001.

[8] L. Michael, "Cognitive reasoning and learning mechanisms," in *CEUR Workshop Proceedings*, 2017, vol. 1895.

[9] P. Maurizio and F. Toni, "Learning Assumption-based Argumentation Frameworks," Sep. 2022, [Online]. Available: http://hdl.handle.net/10044/1/98940.

[10] N. Prentzas, A. Nicolaides, E. Kyriacou, A. Kakas, and C. Pattichis, "Integrating machine learning with symbolic reasoning to build an explainable ai model for stroke prediction," in *Proceedings - 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering, BIBE 2019*, Oct. 2019, pp. 817–821, doi: 10.1109/BIBE.2019.00152.

[11] N. Prentzas, A. Gavrielidou, M. Neophytou, and A. Kakas, "Argumentation-based Explainable Machine Learning (ArgEML): a Real-life Use Case on Gynecological Cancer," in *CEUR Workshop Proceedings*, 2022, vol. 3208.

[12] R. Riveret, S. Tran, and A. D. A. Garcez, "Neural-symbolic probabilistic argumentation machines," in *17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020*, 2020, vol. 2, doi: 10.24963/kr.2020/90.

[13] E. Tsamoura, T. Hospedales, and L. Michael, "Neural-Symbolic Integration: A Compositional Perspective," in *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, 2021, vol. 6A, doi: 10.1609/aaai.v35i6.16639.

[14] N. Sendi, N. Abchiche-Mimouni, and F. Zehraoui, "A new transparent ensemble method based on deep learning," in *Procedia Computer Science*, 2019, vol. 159, doi: 10.1016/j.procs.2019.09.182.

[15] L. Rizzo and L. Longo, "An empirical evaluation of the inferential capacity of defeasible argumentation, non-monotonic fuzzy reasoning and expert systems," *Expert Syst. Appl.*, vol. 147, 2020, doi: 10.1016/j.eswa.2020.113220.

[16] L. Longo, L. Rizzo, and P. Dondio, "Examining the modelling capabilities of defeasible argumentation and non-monotonic fuzzy reasoning," *Knowledge-Based Syst.*, vol. 211, 2021, doi: 10.1016/j.knosys.2020.106514.

[17] L. Rizzo and L. Longo, "A qualitative investigation of the degree of explainability of defeasible argumentation and non-monotonic fuzzy reasoning," in *CEUR Workshop Proceedings*, 2018, vol. 2259.

[18] L. Rizzo, L. Majnaric, and L. Longo, "A Comparative Study of Defeasible Argumentation and Non-monotonic Fuzzy Reasoning for Elderly Survival Prediction Using Biomarkers," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial*

Intelligence and Lecture Notes in Bioinformatics), 2018, vol. 11298 LNAI, doi: 10.1007/978-3-030-03840-3_15.

[19] L. Rizzo and L. Longo, "Comparing and extending the use of defeasible argumentation with quantitative data in real-world contexts," *Inf. Fusion*, vol. 89, 2023, doi: 10.1016/j.inffus.2022.08.025.

[20] N. Prentzas, C. S. Pattichis, and A. Kakas, "Explainable Machine Learning via Argumentation," 2023.

[21] A. C. Kakas, P. Moraitis, and N. I. Spanoudakis, "GORGIAS: Applying argumentation," *Argument Comput.*, vol. 10, no. 1, pp. 55–81, 2019, doi: 10.3233/AAC-181006.

[22] H. Deng, "Interpreting tree ensembles with inTrees," *Int. J. Data Sci. Anal.*, vol. 7, no. 4, pp. 277–287, 2019, doi: 10.1007/s41060-018-0144-8.

[23] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?' Explaining the predictions of any classifier," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, vol. 13-17-Augu, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[24] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017, vol. 2017-Decem, pp. 4766–4775.

[25] M. Setzu, R. Guidotti, A. Monreale, F. Turini, D. Pedreschi, and F. Giannotti, "GLocalX - From Local to Global Explanations of Black Box AI Models," *Artif. Intell.*, vol. 294, 2021, doi: 10.1016/j.artint.2021.103457.