Argue to Learn: Accelerated Argumentation-Based Learning*

H. Ayoobi†, M. Cao‡, R. Verbrugge†, B. Verheij†
†. *Department of Artificial Intelligence, Bernoulli Institute*
‡. *Institute of Engineering and Technology (ENTEG)*
Faculty of Science and Engineering, University of Groningen

***Abstract*—Human agents can acquire knowledge and learn through argumentation. Inspired by this fact, we propose a novel argumentation-based machine learning technique that can be used for online incremental learning scenarios. Existing methods for online incremental learning problems typically do not generalize well from just a few learning instances. Our previous argumentation-based online incremental learning method outperformed state-of-the-art methods in terms of accuracy and learning speed. However, it was neither memory-efficient nor computationally efficient since the algorithm used the power set of the feature values for updating the model. In this paper, we propose an accelerated version of the algorithm, with polynomial instead of exponential complexity, while achieving higher learning accuracy. The proposed method is at least 200× faster than the original argumentation-based learning method and is more memory-efficient.**

***Index Terms*—Argumentation-Based Learning, Online Incremental Learning, Argumentation Theory**

I. Introduction

Argumentation-Based Learning (ABL) [1], [2] outperformed other online incremental learning approaches and was shown to be successful for handling unforeseen failures. However, ABL is not efficient in terms of space and computational complexity. Therefore, the current ABL approach is not usable for high dimensional datasets. In this paper, we propose a novel Accelerated Argumentation-Based incremental online Learning (AABL) method that has a lower space and computational complexity and higher learning accuracy. This entails lower run-time and memory consumption. Moreover, like the original ABL, AABL can generate a set of explainable hypotheses (arguments) for predicting the best recovery behavior (class label).

A. Argumentation in Machine Learning

Argumentation is a reasoning model based on interaction between arguments [3]. Argumentation has been used in various applications such as non-monotonic reasoning [4], inconsistency handling in knowledge bases [5], and decision making [6]. In [7], Dung has defined an Abstract Argumentation Framework (AF) as a pair of the arguments and a attack binary relation among the arguments. Extending Dung's idea, some arguments can support a conclusion and others might be against (attacking) that conclusion in the Bipolar Argumentation Framework (BAF) [8]. According to a survey by Cocarascu et al. [9], the works using argumentation in supervised learning are listed as follows. Argumentation-Based Machine Learning (ABML) [10] uses the CN2 classification approach [11]. This method uses experts' arguments to improve the classification results. The paper by Amgoud et al. [12], [13] explicitly uses argumentation. There are other approaches for improving classification using argumentation in the literature [14].

B. The Expansions

This research is an expansion of our previous paper [1], [2]. The specific expansions are listed as follows.

- Proposing a simpler architecture of the model using only a BAF rather than using both the AF and a BAF.
- Accelerating the prediction and update procedure of the model by introducing an algorithm with lower space and computational complexity by going from exponential to polynomial complexity.
- Including more evaluation scenarios with different levels of complexities.
- Adding run-time and memory usage analysis for both the proposed and previous ABL method.
- Specifying the algorithms in the proposed method by adding pseudocodes to explain argumentation-based learning in more detail.

II. Background

The bipolar argumentation framework [8] is the main building block of our proposed accelerated argumentation-based learning approach. Argumentation-based learning and BAF are formally defined in this section.

A. Argumentation-Based Learning

Using the combination of AF and BAF, argumentation-based learning has been proven to outperform state-of-the-art online incremental learning methods [1], [2]. ABL extracts a set of relevant hypotheses from the learning instances in an online manner and explicitly represents the knowledge acquired from the learning instances as an explainable set of rules as arguments and defeasibilty relations among them. ABL can learn with fewer learning instances. However, it lacks the ability to work with higher dimensional data since it uses all the subsets of the feature values as the supporting nodes and this makes the model slow and not memory-efficient. In this paper, we will propose a new argumentation-based learning approach which resolves these issues.

* This work is conducted at DSSCand sponsored by a Marie Skłodowska-Curie COFUND grant, agreement no. 754315.

978-1-6654-4337-1/21/$31.00 ©2021 IEEE
DOI 10.1109/ICMLA52953.2021.00183

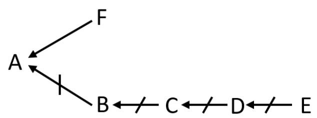


Fig. 1. A Bipolar Argumentation Framework (BAF).

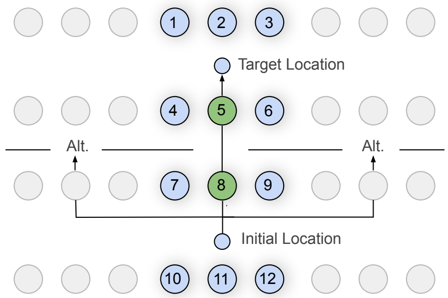


Fig. 2. Schematic overview of the possible failure state scenarios. Only the green location is relevant for finding the best recovery behavior. Alt. stands for the Alternative Route recovery behavior.

B. Formal Definition of an Abstract Bipolar Argumentation Framework

An Abstract Bipolar Argumentation Framework (*BAF*) [8] is a triple of the form $< AR, R_{att}, R_{sup} >$ where *AR* is the finite set of arguments, $R_{att} \subseteq AR \times AR$ is the *attack* set and $R_{sup} \subseteq AR \times AR$ is the *support* set. Considering A_i and $A_j \in AR$, then A_i R_{att} A_j means that A_i attacks A_j and A_i R_{sup} A_j means that A_i supports the argument A_j.

The semantics of BAF are as follows:

(**Conflict-Free**) Let $S \subseteq AR$. S is conflict-free iff there is no $B, C \in S$ such that B attacks C.

(**Admissible set**) Let $S \subseteq AR$. S is admissible iff S is conflict-free, closed for R_{sup} (if $B \in S$ and B R_{sup} $C \Rightarrow C \in S$) and S defends all its elements.

(**Preferred extension**) The set $E \subseteq AR$ is a preferred extension iff E is inclusion-maximal among the admissible sets. An inclusion-maximal set among a collection of sets is a set that is not a subset of any other set in that collection.

Figure 1 shows a bipolar argumentation framework. The admissible sets are {}, {E}, {A, C, E}, {A, C, E, F}. The preferred extension in this BAF is {A, C, E, F}.

C. Online Incremental Machine Learning Algorithms

A recent study on the comparison of the state-of-the-art methods for incremental online machine learning [15] shows that Incremental Support Vector Machines (*ISVM*) [16], [17] together with LASVM [18], which is an online approximate SVM solver, and Online Random Forest (*ORF*) [19] outperform the other methods. The original ABL approach outperformed all these methods in terms of accuracy and learning speed [1], [2]. Therefore, we only compare the proposed AABL method with the original ABL approach. Both AABL and ABL can be utilized in open-ended learning scenarios [20].

III. Scenarios

The performance of the different methods is tested using three test scenarios. The aim of these test scenarios is to model a situation where a programmer has provided an initial solution (e.g., a top level behavior such as entering the room), while he has not accounted for all possible failures (e.g., objects and persons blocking the entrance), allowing, however, the robot to find new solutions whenever a (previously unseen) failure occurs.

The basic setup of the test scenarios is illustrated in Fig. 2. The high-level behavior of the robot aims to proceed from the initial location to the target location using three entrances. Different obstacles might be on its way to the target location. Looking at all the obstacle locations at once, the robot can reach the goal by choosing the best recovery behavior.

A. Recovery Behaviors

Whenever the robot is confronted with an obstacle, it may use any of the following recovery behaviors to resolve the issue:

- Continue: This solution is only useful if the failure has resolved itself (e.g., the obstacle moved away).
- Push: The robot can try pushing any obstacle.
- Ask: The robot can try to ask any obstacle to move.
- Alternative Route (Alt): The robot can move to another entrance to reach the target location.

It is important to note that choosing Alternative Route as the best recovery behavior may not always lead to success, because the robot may again be confronted with new obstacles (Fig. 2). Moreover, the best recovery behavior depends on the type, color and location of the obstacle.

B. Test Scenario 1

In this scenario, three concepts (ball, box or person) with four colors (red, blue, green or yellow) can be presented in one of the locations 1 to 6 (Fig. 2). Locations 7 to 12 play no role in this scenario. There can be either zero or one combination of color-concept in each location. Only location number 5, marked in green (Fig. 2), is relevant for choosing the best recovery behavior. It is important to notice that the robot does not know this fact and it should infer it by itself.The number of possible combinations of the color-concept in each location is 13 (3 types $\times$ 4 colors + "no obstacle" = 13). Since there are 6 locations in this scenario, the number of all possible states in this scenario is $13^6 = 4,826,809$.

Notice that colors can have meaningful interpretations. For instance, the red object might be heavy and cannot be pushed, while green ones are light. Using the colors instead of these realistic features simplifies the scenarios with fewer features.

C. Test Scenario 2

This scenario is more complex than the first scenario, since each color-concept combination can be presented in one of the locations 1 to 9 (Fig. 2). Here, only the green locations 5 and 8 are relevant for determining the best recovery behavior. The number of all possible states in this scenario is $13^9 = 10,604,499,373$.

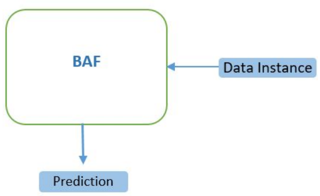


Fig. 3. Architecture of the proposed Argumentation-based learning approach.

Order	Color	Concept	Best Recovery Behavior
1	Red	Ball	Push
2	Red	Box	Alternative Route
3	Red	Person	Ask
4	Green	Ball	Push
5	Green	Box	Alternative Route
6	Green	Person	Ask
7	Blue	Ball	Push
8	Blue	Box	Alternative Route
9	Blue	Person	Alternative Route
10	Yellow	Ball	Push
11	Yellow	Box	Alternative Route
12	Yellow	Person	Ask
13	None	None	Continue

TABLE I
POSSIBLE COMBINATIONS OF COLOR-TYPE WITH THE BEST RECOVERY BEHAVIORS

D. Test Scenario 3

The third scenario is the most complex scenario in all the scenarios. Each color-type combination can be presented in any of the twelve different locations (Fig. 2). Like the previous scenario, only locations number 5 and 8 play a role for determining the best recovery behavior. In this Scenario, the number of all the possible states is $13^{13} = 302,875,106,592,253$.

IV. METHOD

In this section, with an illustrating example, we first explain the Bipolar Argumentation Framework (BAF) unit which is the main building block of the proposed approach. Subsequently, we define AABL, and its updating procedure.

A. Explanation of the Method with an Illustrating Example

We first use the simplified version of the test scenarios with only one location ahead of the agent (instead of 6, 9 or 12 locations). Figure 3 shows the architecture of the proposed argumentation-based learning approach.

Using the randomly generated Table I, the robot is initially confronted with a Red-Ball (R-Ba) and tries different recovery behaviors to find out that the best choice is *Push*. The model initially gets updated by the subsets of feature values with length 1 ($L := 1$). This means that the supporting nodes R and Ba are added to the *Push* recovery behavior (Fig. 4). Subsequently, the agent is confronted with a Red-Box (R-Bo) for which the subsets of feature values with length $L = 1$ consist of R and Bo. Looking at the current state of the BAF, R supports the *Push* recovery behavior and it is chosen as the model's prediction. Since, this is a wrong choice, the agent try other recovery behaviors and find "Alternative route" (*Alt*) as the best recovery behavior. Therefore, the *Alt* node gets updated with its supporting nodes R and Bo and also a bidirectional attack among *Alt* and *Push* nodes. Since R supports both *Push* and *Alt* recovery behaviors, it is not a unique supporter for each of them and it will be pruned from

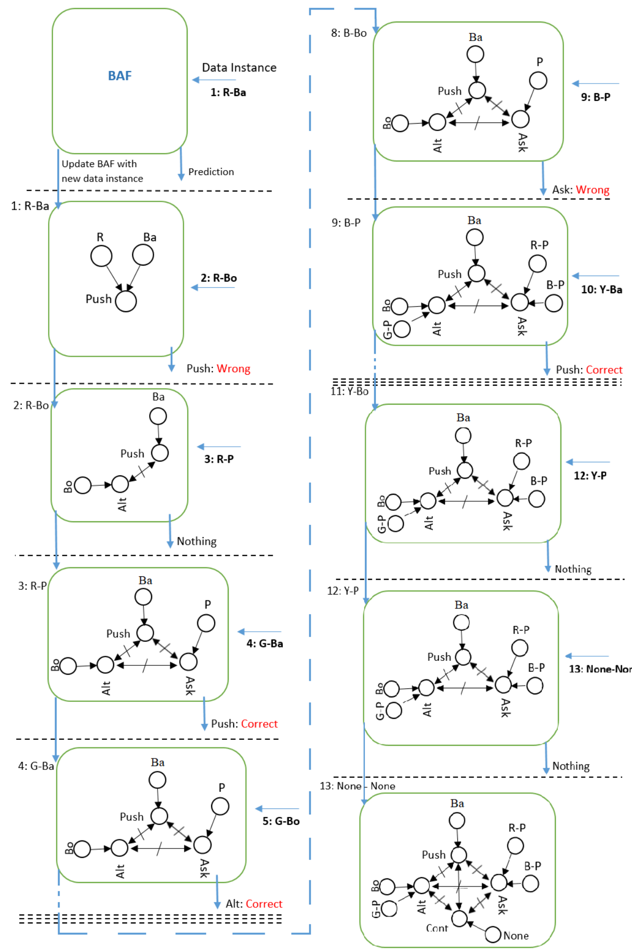


Fig. 4. Example of Argumentation-Based Learning for the illustrating example. First part

both the recovery behaviors and will be marked as a node which can no longer support any recovery behavior nodes in the future.

For the third learning instance the robot is confronted with a Red-Person (R-P) and the models does not have any prediction since no current recovery behavior node in the BAF has either P or R in its supporting nodes. The BAF unit gets updated with only P as a supporting node for the *Ask* since R has been previously marked as a non-supporting node and bidirectional attack relations are added among all pairs of the recovery behaviors. Subsequently, the agent is confronted with a Green-Ball (G-Ba) obstacle and since Ba supports the *Push* in the BAF unit, *Push* is chosen as a prediction for the best recovery behavior. The BAF gets updated using G supporting node for *Push* recovery behavior. This process will continue and the model predicts the best recovery behavior correctly in the subsequent obstacle confrontations until the agent is confronted with a blue person.

When the agent is confronted with a Blue-Person (B-P), it chooses *Ask* as the best recovery behavior. This is a reasonable choice because *Ask* was the best recovery behavior for all the previous cases where a *Person* was the obstacle, namely, in R-P and G-P. However, it turns out that the best choice for B-P is *Alt*. Since, the subsets of feature values with length 1 were not adequate for choosing the best recovery behavior for both the *Ask* and the *Alt* recovery behaviors, the length of the subsets of the feature values is incremented i.e. $L := L+1$ in this case $L = 2$. Therefore, the supporting node B-P is added to the *Alt* recovery behavior and the supporting nodes R-P and G-P are added to the *Ask* recovery behavior while *P* is pruned and marked as a non-supportable node. The model predicts the correct categories for the rest of the learning instances until it is confronted with a Yellow-Person (Y-P). In this case the model does not have any guess for the best recovery behavior and gets updated with the Y-P supporting node. The last learning instance None-None is a new case based on the previous agent's experiences and it does not have any prediction for that. Finally the model gets updated and a new recovery behavior node *Continue* (*Cont*) is added to the BAF model. In this illustrating example, the proposed approach has seven correct predictions and two wrong predictions out of all the thirteen instances while having no other predictions for the other four cases.

Comparing the number of nodes between the proposed model and the previous argumentation-based learning model [1], [2], the number of saved nodes in memory for our proposed approach decreases from 44 to 11 and the number of attack or support relations decreases from 40 to 13 (Fig. 5). Since each attack or support relation is between two nodes, the memory usage for these relations is twice the memory usage of the saving nodes. Moreover, the supporting weights and argument weights are reduced from 40 to 0. Therefore, the total memory usage decreases from $44+(40\times 2)+40 = 164$ to $11+(13\times 2)+0 = 37$ which is more than four times (4×) lower in this small illustrating example. Moreover, the lower number of saved nodes in the memory results in the lower number of comparisons between the feature values and lower ultimate run-time.

B. Accelerated Argumentation-Based Learning

As explained in the previous sub-section, the main difference among the proposed Accelerated Argumentation-Based Learning (AABL) approach and the ABL is the architecture of the model. Here, only a the BAF unit is utilized. Algorithm IV-B shows the pseudocode of the proposed approach for AABL. Instead of initiating the model with all the subsets of feature values, the model begins with the subsets of length 1 and increases the length of the subsets when needed. This way the number of required computations in the algorithm is significantly reduced. Using the extracted subsets and the set of supporting nodes in the model, the best recovery behavior (action) is predicted. If the model could predict a recovery behavior in the previous step, it will be applied to the environment. If there exits more than one predictions, one of them is chosen randomly. Otherwise, the random choice is among all the possible recovery behaviors (actions). Subsequently, the BAF unit is updated using the algorithm IV-B. The updating process has two steps, namely, updating the attack relations and updating the support relations. When a new recovery behavior (action) is added to the model, bidirectional attacks will be added between the newly added recovery behavior (action) and all the other previous recovery behaviors (actions) in the model. The supporting nodes are then added or pruned based on their uniqueness in supporting a recovery behavior (action). When all the supporting nodes are pruned, the length of the subsets of the feature values will increase.

	Proposed Approach (secs)	Original ABL (secs)
First Scenario	0.42	84.76
Second Scenario	3.16	3318.68
Third Scenario	13.56	87088.60

TABLE II
COMPARISON OF RUN-TIMES IN SECONDS FOR DIFFERENT SCENARIOS.

V. EXPERIMENTS

A. Experimental Setup

In all the experiments, a table like Table I is randomly generated. Using this randomly generated table, we then randomly generate the three scenarios as explained in Section III. Each experiment has been conducted ten times (iterations) and the average result is reported. In order to compare the accuracy of both the proposed AABL and the original version of ABL [1], [2], we have set the limit of 200 recovery attempts at each iteration. The run-times are reported in seconds.

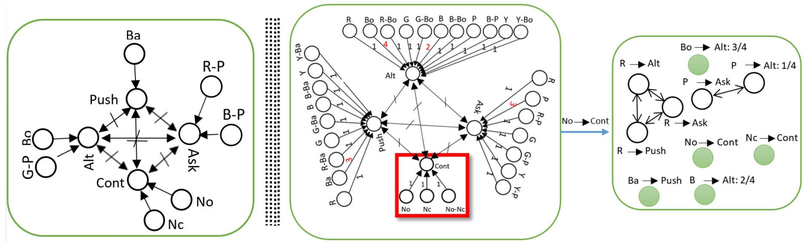
B. Experimental Results

In this section, three sets of experiments have been conducted. First, the run-time of both approaches for all the scenarios have been compared and the computational complexity of the approaches has been discussed. Second, the memory consumption of both methods has been compared. Moreover, the space complexity of the algorithms is computed. Third, the learning accuracy of both approaches has been compared.

1) Comparison of Run-times: Table II shows the comparison of the run-times of the proposed AABL and the original ABL over these scenarios. As you can see, the newly proposed approach outperforms the original version of ABL by a large margin. For the first scenario, the run-time of the original ABL is 84.76s, while the newly proposed method has the run-time of 0.42s. This means that AABL runs almost 200 *times* faster than ABL. The run-time of our proposed approach for the second scenario is 3.16s, while it is 3318.68s for the original ABL approach. This means that the proposed approach is more than 10^3 *times* faster than the previous approach for the second scenario. For the third scenario, the proposed ABL algorithm runs 6×10^3 *times* faster than the original ABL algorithm.

Figure 6 shows the relation between the number of locations in a scenario like the first scenario and the run-time of different approaches. The run-time of the original ABL approach exponentially increases while the proposed accelerated ABL approach maintains much lower run-time that is linearly dependent on the number of locations. Since the first scenario is only dependent on the location number 5, only the subsets with the length 1 from the feature values are extracted.

2) Computational Time Complexity: In the previous section, we have explained that the proposed ABL method begins with extracting the subsets of feature values with length 1 and then increases the length of subsets if needed. Moreover, pruning the unnecessary supporting nodes reduces the required



a) Our Approach
b) Original ABL approach [1], [2]

Fig. 5. Comparison of the ABL models after training on the illustrating example

Algorithm 1: Argumentation-Based Learning

input: Current **BAF** graph, Data Instance **X** entering the argumentation-based learning model, feature values subsets' length **L**, The class label or **B**est **R**ecovery **B**ehavior (**BRB**) for **X**
output: The predicted label for **X** called **Y**

Start:
- Extract all feature value combinations in **X** with length **L** and add them to a list called **Combs**.
- Let **SNs** be the set of supporting nodes (in form of "supporting-node → supported-node") in the BAF.

for *(all **sn** in **SNs**)* **do**
    for *(all **comb** in **Combs**)* **do**
        if *(sn.supporting-node==comb)* **then**
            Y.Add(sn.supported-node)

if *(**Y** is not empty)* **then**
    if *(Length(**Y**)==1)* **then**
        - Apply **Y** to environment and observe the **result**.
    else
        - Select a prediction in **Y** at random (Y := Y[random_index]) and observe the **result**.
else
    - Randomly choose a prediction from the available class labels (observed recovery behaviors).

- **should_Increment_L** := Update the **BAF** unit (using Algorithm 2 with input parameters: current **BAF** graph, **BRB**, **Combs**, **SNs**).

while *(**should_Increment_L** == True)* **do**
    - L := L+1
    - Compute the combinations of the feature values again as **Combs**.
    - **should_Increment_L** := Update the model with Algorithm 2.

return Y

	Proposed Approach (MBs)	Original ABL (MBs)
First Scenario	0.9	20.4
Second Scenario	1.3	161.1
Third Scenario	1.7	392.73

TABLE III
COMPARISON OF MEMORY USAGE FOR DIFFERENT SCENARIOS.

number of computations of the approach. Assuming that n is the number of features in a dataset, for the first scenario the order of the proposed algorithm is $O(n)$, while the original ABL method has $O(2^n)$ since it extracts all the subsets of the feature values. The reason that the proposed algorithm is in $O(n)$ for the first scenario is that in the worst case it only uses the subsets of feature values with length 1. For the second and third scenarios, the algorithm uses subsets of length 2 in the worst case since there are only two relevant locations in their state space. Therefore, the order of the proposed algorithm is $O(n^2)$ while it is $O(2^n)$ for the original ABL.

3) Comparison of the Memory Usage: In order to compare the memory usage of both the previous ABL approach and the new proposed ABL approach, we have made a comparison. Table III shows the comparison of the memory usage of both approaches in MBs for all the scenarios. For the first scenario,

Algorithm 2: Updating the BAF Unit

input: Current **BAF** graph, class label (**B**est **R**ecovery **B**ehavior) **BRB**, Combinations of feature values for **X** called **Combs**, Set of **S**upporting **N**odes in the BAF graph **SNs**
output: A Boolean variable "**should_Increment_L**" that tells whether **L** needs to be incremented or not.

Start:
- Let **RNs** be the set of all the class labels (**R**ecovery behavior **N**odes) in the BAF.
- Let **attacks** be the set all the attack relations (for **a** ∈ **RNs** and **b** ∈ **RNs** the attack relations are in form of "**a** → **b**") among the class labels (recovery behavior nodes) in the BAF.

Step 1: (Updating attack relations and class labels)

if *(**BRB** is not in BAF)* **then**
    - add **BRB** to the BAF graph;
    - add bidirectional attacks between **BRB** and all the other class labels (recovery behavior nodes) as follows:
    for *all **rn** in **RNs*** **do**
        - **attacks.add(BRB → rn)**
        - **attacks.add(rn → BRB)**

Step 2: (Updating support relations)

- Let **should_Increment_L := True**

for *(all **comb** in **Combs**)* **do**
    - Let **Add_Support :=True**
    for *(all **sn** in **SNs**)* **do**
        if *(**sn**.supporting-node == **combs**)* **then**
            - **Add_Support :=False**
            if *(**sn**.supported-node ≠ **BRB**)* **then**
                - Mark **comb** as a non-unique node that can not support any node in future.
                - Remove **sn** from the set supporting nodes in BAF **SNs**.
    if *(**comb** is not Marked)* **then**
        - **should_Increment_L := False**
        if *(**Add_Support** == True)* **then**
            - **SNs**.add(**comb** → **BRB**)

return should_Increment_L

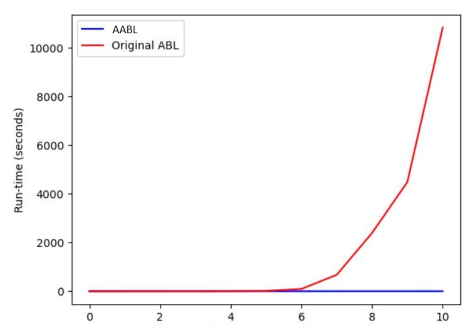


Fig. 6. The relation between the run-time and the number of locations in a scenario like the first scenario.

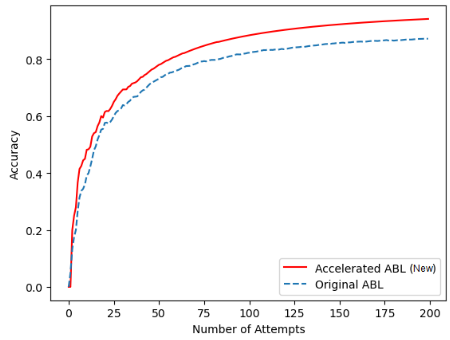


Fig. 7. The comparison of learning accuracy vs number of recovery attempts between the proposed AABL and the original ABL for the first scenario.

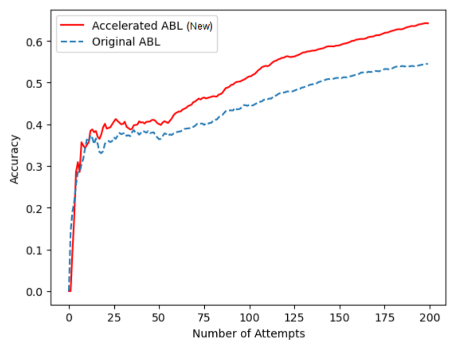


Fig. 8. The comparison of learning accuracy vs number of recovery attempts between the proposed AABL and the original ABL for the second scenario.

our current method has more than 20 *times* lower memory consumption. Moreover, for the second scenario, the memory consumption is more than 70 *times* reduced in the newly proposed approach. The proposed approach uses 200 *times* lower memory for the third scenario.

4) Space Complexity: Space complexity of the proposed AABL method is directly related to the computational time complexity of the approach. Therefore, for the first scenario, the space complexity of the proposed algorithm is $O(n)$ where n is the number of features in the dataset. The space complexity of the proposed approach for the second and third scenarios is $O(n^2)$. In all the scenarios, the space complexity of the original ABL approach is $O(2^n)$.

5) Evaluating the Accuracy: In order to evaluate the performance of both methods based on the learning accuracy, we have conducted two experiments. Figure 7 compares the accuracy of both methods for the first scenario. The comparison of both methods for the second scenario is illustrated in Figure 8. In both cases, the proposed ABL approach has higher accuracy.

VI. Conclusion

Argumentation-Based online incremental Learning (ABL) has been introduced recently in [1], [2]. ABL outperformed other state-of-the-art online incremental learning algorithms. Although ABL has higher learning accuracy than other approaches, the it is not suitable for high dimensional problems. The reason lies in the high computational complexity of the approach. In this paper, we have proposed Accelerated Argumentation-Based Learning (AABL), which has lower computational complexity, and memory usage. The resulting approach can be used for higher dimensional problems while having a better learning accuracy than the original version of the ABL algorithm. We have conducted three sets of experiments with more complex scenarios and analyzed the run-time and memory usage of the methods. The proposed approach outperforms the original version of ABL algorithm in terms of run-time and memory usage by a large margin while slightly outperforming the original ABL in terms of accuracy. The lower computational complexity of the proposed approach makes it applicable for wider range of machine learning problems.

References

[1] H. Ayoobi, M. Cao, R. Verbrugge, and B. Verheij, “Handling unforeseen failure conditions using argumentation-based learning,” in *International Conference on Automation Science and Engineering*, IEEE, 2019.

[2] H. Ayoobi, M. Cao, R. Verbrugge, and B. Verheij, “Argumentation-based online incremental learning,” *IEEE Transactions on Automation Science and Engineering (TASE)*, 2021.

[3] F. H. Van Eemeren, B. Garssen, E. C. Krabbe, A. F. S. Henkemans, B. Verheij, and J. H. Wagemans, *Handbook of Argumentation Theory*. Dordrecht: Springer, 2014.

[4] L. Rizzo and L. Longo, “An empirical evaluation of the inferential capacity of defeasible argumentation, non-monotonic fuzzy reasoning and expert systems,” *Expert Systems with Applications*, vol. 147, p. 113220, 2020.

[5] A. Vassiliades, N. Bassiliades, and T. Patkos, “Argumentation and explainable artificial intelligence: a survey,” *The Knowledge Engineering Review*, vol. 36, 2021.

[6] K. Atkinson, T. Bench-Capon, and D. Bollegala, “Explanation in AI and law: Past, present and future,” *Artificial Intelligence*, vol. 289, p. 103387, 2020.

[7] P. M. Dung, “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games,” *Artificial Intelligence*, vol. 77, no. 2, pp. 321–357, 1995.

[8] L. Amgoud, C. Cayrol, M.-C. Lagasquie-Schiex, and P. Livet, “On bipolarity in argumentation frameworks,” *International Journal of Intelligent Systems*, vol. 23, no. 10, pp. 1062–1093, 2008.

[9] O. Cocarascu and F. Toni, “Argumentation for machine learning: A survey,” in *COMMA*, pp. 219–230, 2016.

[10] M. Mozina, J. Zabkar, and I. Bratko, “Argument based machine learning,” *Artificial Intelligence*, vol. 171, no. 10-15, pp. 922–937, 2007.

[11] P. Clark and T. Niblett, “The CN2 Induction Algorithm,” *Machine Learning*, vol. 3, no. 4, pp. 261–283, 1989.

[12] L. Amgoud and M. Serrurier, “Agents that argue and explain classifications,” *Autonomous Agents and Multi-Agent Systems*, vol. 16, no. 2, pp. 187–209, 2008.

[13] O. Cocarascu, A. Stylianou, K. Čyras, and F. Toni, “Data-empowered argumentation for dialectically explainable predictions,” in *ECAI 2020*, pp. 2449–2456, IOS Press, 2020.

[14] L. Carstens and F. Toni, “Using argumentation to improve classification in natural language problems,” *ACM Transactions on Internet Technology (TOIT)*, vol. 17, no. 3, p. 30, 2017.

[15] V. Losing, B. Hammer, and H. Wersing, “Incremental on-line learning: A review and comparison of state of the art algorithms,” *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.

[16] A. Soula, K. Tbarki, R. Ksantini, S. B. Said, and Z. Lachiri, “A novel incremental Kernel Nonparametric SVM model (iKN-SVM) for data classification: An application to face detection,” *Engineering Applications of Artificial Intelligence*, vol. 89, p. 103468, 2020.

[17] R. Kashef, “A boosted SVM classifier trained by incremental learning and decremental unlearning approach,” *Expert Systems with Applications*, vol. 167, p. 114154, 2021.

[18] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, “Fast kernel classifiers with online and active learning,” *Journal of Machine Learning Research*, vol. 6, no. Sep, pp. 1579–1619, 2005.

[19] G. Biau and E. Scornet, “A random forest guided tour,” *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[20] H. Ayoobi, M. Cao, R. Verbrugge, and B. Verheij, “Local-HDP: Interactive Open-ended 3D Object Category Recognition in Real-Time Robotic Scenarios,” *Robotics and Autonomous Systems*, 2021.