Working Notes of the 1995 AAAI Fall Symposium on Active Learning, 70–73
Boston, MA, November 10-12, 1995

Learning Evolving Concepts Using Partial-Memory Approach

Marcus A. Maloof and Ryszard S. Michalski*

Machine Learning and Inference Laboratory
George Mason University
Fairfax, VA 22030
{maloof, michalski}@aic.gmu.edu
*Also with the Institute of Computer Science, Polish Academy of Sciences

Abstract

This paper addresses the problem of learning *evolving concepts*, that is, concepts whose meaning gradually evolves in time. Solving this problem is important to many applications, for example, building intelligent agents for helping users in Internet search, active vision, automatically updating knowledge-bases, or acquiring profiles of users of telecommunication networks. Requirements for a learning architecture supporting such applications include the ability to incrementally modify concept definitions to accommodate new information, fast learning and recognition rates, low memory needs, and the understandability of computer-created concept descriptions. To address these requirements, we propose a learning architecture based on Variable-Valued Logic, the Star Methodology, and the AQ algorithm. The method uses a partial-memory approach, which means that in each step of learning, the system remembers the current concept descriptions and specially selected representative examples from the past experience. The developed method has been experimentally applied to the problem of computer system intrusion detection. The results show significant advantages of the method in learning speed and memory requirements with only slight decreases in predictive accuracy and concept simplicity when compared to traditional batch-style learning in which all training examples are provided at once.

Introduction

This paper addresses the issue of learning evolving concepts, that is, concepts whose meaning is gradually evolving in time. This issue is closely related to the issue of incremental concept learning, in which the concept to be learned remains constant, but training examples are incrementally supplied. The system uses these examples to improve its currently held concept definitions. In learning evolving concepts, the concept itself is changing, and examples are used for capturing these changes in concept meaning.

This issue is important for a variety of applications, such as intelligent software agents, active vision systems, and computer intrusion detection systems. Many of the characteristics of these applications are similar in that training data is distributed over time, the concepts in these applications often change, and the system must operate semi-autonomously or autonomously during periods when feedback from the user or the environment is unavailable. It is also important that the system is able to learn and recognize concepts quickly and employs easy to understand symbolic concept descriptions.

To address these requirements, we have developed a novel incremental learning methodology based on Variable-Valued Logic (Michalski 1973), the Star Methodology (Michalski 1983), and the AQ algorithm (Michalski 1969). The method operates in a partial-memory mode (Reinke & Michalski 1988) in which the system maintains a set of representative examples derived from past experience. A training example is considered representative if it expands or constrains concepts in the event space. The methodology also consists of aging and forgetting mechanisms for managing representative examples, inductive support mechanisms for uncertainty management, and consistency maintenance routines for governing the system during periods of autonomous functioning. Experiments have been conducting using the dynamic knowledge-based application of computer intrusion detection. Results demonstrate that partial-memory incremental learning yielded significant improvements in learning time and memory requirements at the cost of slightly lower predictive accuracy and slightly more complex concept descriptions when compared to traditional batch learning in which all examples are provided at once (Maloof & Michalski 1995).

Background

In conventional learning methods, concepts are assumed to be constant, that is, their inherent meaning does not change. The goal of the learner is to capture this meaning by observing concept examples, which can be given at once (batch learning) and incrementally. This paradigm works well for knowledge-based system applications which do not change in time.

Applications such as dynamic knowledge-bases, intelligent agents, and active vision systems violate many of the traditional assumptions of concept learning. Concepts are not static; they evolve over time. All training examples are not available at any given time; training examples are distributed over time.

Consequently, the system must not only learn over time, but it must also learn a changing concept.

Incremental learning is inherently a temporal process. Incremental learning can be conducted using either an evolutionary or revolutionary scheme (Michalski 1985). With an evolutionary scheme, existing knowledge is modified based on new training examples. STAGGER (Schlimmer 1987) is an incremental learning systems that takes an evolutionary approach. With a revolutionary approach, old knowledge is discarded and new knowledge is learned from the given training examples. AQ15 (Hong et al. 1986) is an example of an incremental learning system that takes a revolutionary approach. Finally, a hybrid approach takes elements from both the revolutionary and evolutionary approaches. CAP (Mitchell et al. 1994), for example, learns new decision rules from training data and incorporates these new rules into the existing knowledge-base. CAP is hybrid system because it learns new knowledge solely from training examples (i.e., a revolutionary approach) and then incorporates this new knowledge into the existing knowledge-base (i.e., an evolutionary approach).

Incremental learning systems can work in one of three different modes: *no-memory*, *partial-memory*, or *full-memory*. In the no-memory mode, the system does not use any past training examples for modifying or updating the currently held hypotheses. STAGGER (Schlimmer 1987) uses a no-memory mode. In a partial-memory mode, a subset of all previously seen training examples is maintained and used for subsequent learning. The work presented here uses a partial-memory mode. Aha and Kibler (1992) also use a partial-memory mode, but for instance-based learning which does not form generalized concept descriptions. Finally, under a full-memory mode, all past training examples are maintained and used in the process of modifying existing hypothesis. AQ15 (Reinke & Michalski 1988, Hong et al. 1986) operates using a full-memory mode.

Learning evolving concepts adds another layer of difficulty to the process of incremental learning. Concepts can no longer be assumed to be constant. This means that after some time, previously seen training examples may cease to be correct, since the concept could have evolved. Some previously positive examples may become negative and vice versa. A system for learning evolving concepts must be able to cope with such situations. The main idea for solving this problem is to pay less attention to the past examples and more attention to the newer concept examples in the process of updating concept descriptions.

Incremental learning, especially learning of evolving concepts, plays an important role in such applications such as those mentioned previously. These applications require constant interaction with the environment or a user, and it is therefore impossible to provide the system will all of the *a priori* knowledge it needs. Furthermore, the user's needs or the characteristics of the environment change over time. Lastly, there will likely be periods of time when the system must function autonomously and will not have the benefit of user or environmental feedback.

Proposed Method

A general flow diagram for the proposed method is shown in Figure 1. The method has two phases of operation: the *start-up* phase and the *update* phase. In the start-up phase the system is provided enough initial knowledge to function in the given environment.

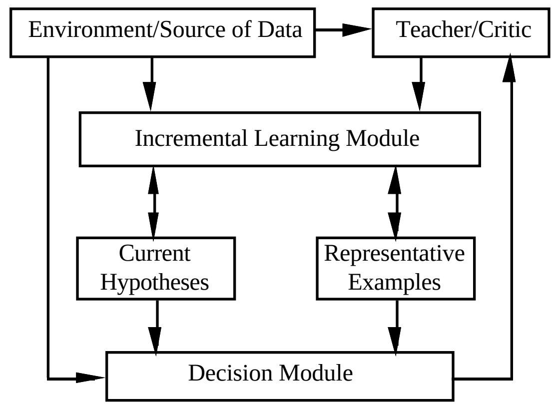

Figure 1: Architecture for partial-memory incremental learning.

This phase can be accomplished by directly introducing the required start-up knowledge to the system or through learning from examples. In the latter case, a teacher selects initial training examples from a source of data or the environment, and represents them in a pre-defined representation space (defined by attributes or measurements applied to the objects). These examples are used by a learning system to form the initial knowledge.

In the update phase, the system is placed in application environment, and further learning is based on the feedback from the environment or a teacher. In this phase, the system receives unclassified observations and assigns them decisions based on the rules (current hypotheses) in its knowledge base. The decisions are evaluated by the teacher (or the environment) and when there is any discrepancy between teacher's and system's decisions, the system updates its knowledge base.

The key aspects of the proposed methodology are:

1. employment of an incremental learning process using a partial-memory mode,
2. aging and forgetting mechanisms for representative examples,
3. inductive learning mechanism,
4. consistency maintenance mechanisms, and

5. symbolic concept representation.

The following subsections discuss these key aspects in more detail

An Algorithm for Incremental Learning using a Partial-Memory Mode

Below is an outline of the partial-memory incremental learning algorithm:

Algorithm 1

Given data sets DATA_i, for $i = 1..\infty$

0. $i = 1$
1. $\text{TRAINING\_SET}_i = \text{DATA}_i$
2. $\text{CONCEPTS}_i = \text{Learn}(\text{TRAINING\_SET}_i)$
3. $\text{REPRESENTATIVE}_i = \text{FindRepresentativeExamps}(\text{CONCEPTS}_i, \text{TRAINING\_SET}_i \alpha)$
4. $\text{MISSED}_i = \text{FindMissedNewExamples}(\text{CONCEPTS}_i, \text{DATA}_{i+1})$
5. $\text{TRAINING\_SET}_{i+1} = \text{REPRESENTATIVE}_i \cup \text{MISSED}_i$
6. $i = i + 1$
7. go to step 2

The counter i is a temporal counter that represents the passage of time in the system's environment. Time is assumed to be linear and discrete. When $i = 1$, steps 1–3 relate to the start-up phase. A teacher defines a representation space and collects the initial data set. This should provide enough initial knowledge to the system for it to cope in its intended environment. After learning, the system uses its induced concepts to find a set of past representative examples.

The key issue with partial-memory learning is how to choose the representative past examples. Referring to Figure 2, assume that for some representation (event) space E, we have learned a concept c from a set of training examples. For this work, representative training examples are determined on the basis of the learned concept descriptions (rules in our case) and past training examples. We select representative examples as those that lie at the boundaries of the learned concept descriptions (represented in Figure 2 by the outlined pluses and minuses). All other training examples are discarded.

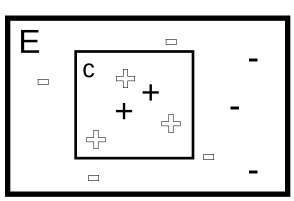

Figure 2: Representative examples in an event space.

Referring again to Algorithm 1, once step 3 has been executed for $i = 1$, the algorithm enters the update phase. At each time step denoted by i, the system receives a non-empty set of observations coming from a pre-defined representation space. Some of these observations will be accompanied by feedback from the teacher denoting their assigned class labels. Misclassified observations signify either incomplete or inconsistent system knowledge, or noise in the data. Assuming the former case, incremental learning is triggered. The system's knowledge is updated using past concepts, representative examples, and misclassified observations. This process repeats for the life of the system.

Aging and Forgetting Mechanisms

Once a partial-memory approach to incremental learning is taken, several important issues arise that are not pertinent to using no or full memory modes. One such issue is memory management of representative examples.

When learning evolving concepts, examples that can be considered *representative* are likely to change over time. Therefore, newer representative examples will tend to be more important than older representative examples.

The set REPRESENTATIVE_i specifies the past examples selected for incremental learning, and a *forgetting function* α specifies the dependence of the importance of the training examples on the time when they were seen. The forgetting function α is constant or monotonically decreases with time. A value of 1 indicates full importantance, while 0 indicates no importance. If α is a constant function equal 1, then we have a conventional incremental learning process of stable, non-evolving concepts. In such a process, both past and new examples are of equal importance. The function α can be selected by an expert or can be learned during the start-up and update phases.

Inductive Support Mechanisms

Another type of weight associated with representative examples and inductive hypotheses, different from that described in the aging process, is the type associated with inductive support. For instance, if over time, the system sees a particular observation repeatedly, then inductive support mechanisms may increase the weight of either the representative example representing this observation, if one is memorized, or the hypothesis that covers the observation. These weights are used in conjunction by the aging and forgetting mechanisms, as well as by the inductive learning process and rule matching routines.

Consistency Maintenance Mechanisms

The system is designed to operate semi-autonomously, when there is little or no feedback from the environment or the user. During these periods, the system simply provides classifications for observations. When feedback on the system's decision becomes available, it may be discovered that some of its past classifications were incorrect. The system can deal with this situation in a variety of ways. The past classifications can simply be ignored. Another possibility is to reclassify them on the basis of the new information. Or, the system can allow the user to reclassify the past observations manually. Note that when we say 'past observations', we mean those observations that were made since the last user feedback.

Symbolic Concept Representations

The final key aspect of this method is the use of a symbolic concept representation, which is important for several reasons. Paramount of these reasons is the need for human comprehension of learned knowledge. Humans must be able to inspect and modify learned knowledge to validate and optimize system performance. This is especially true for situations in which an intelligent system is making decisions that affect other humans.

Conclusions

This paper has discussed a novel incremental learning methodology employing a partial-memory approach. The methodology is designed to support applications such as dynamic knowledge-based systems, intelligent agents, and active vision systems. These applications have similar characteristics in that training data is distributed over time, concepts in these domains often change, and the learning system must function semi-autonomously when feedback from the environment or the user is unavailable.

To cope with these requirements, the methodology incorporates components such as a partial-memory mode based on representative examples, aging and forgetting mechanisms, inductive support mechanisms, and consistency maintenance mechanisms. Among main advantages of the method is the high comprehensibility of the knowledge representation used and the employment of a symbolic learning approach that allows the system to easily modify the hypothesis. Current work is, however, at an early stage of development and many ideas are implemented only in a very rudimentary fashion.

Acknowledgments

The authors thank Eric Bloedorn and Ken Kaufman for many discussions surrounding this work providing insights that were helpful for implementing these ideas. The authors greatly appreciate the support of a Doctoral Fellowship awarded to the first author from the School of Information and Technology Engineering at George Mason University. This research was conducted in the Machine Learning and Inference Laboratory at George Mason University. The Laboratory's research is supported in part by the Advanced Research Projects Agency under Grant No. N00014-91-J-1854, administered by the Office of Naval Research, and the Grant No. F49620-92-J-0549, administered by the Air Force Office of Scientific Research, in part by the Office of Naval Research under Grant No. N00014-91-J-1351, and in part by the National Science Foundation under Grants No. IRI-9020266 and DMI-9496192.

References

Aha, D. W.; Kibler, D.; and, Albert, M. K. 1991. Instance-based learning algorithms. *Machine Learning* 6: 37–66.

Hong, J.; Mozetic, I.; and Michalski, R. S. 1986. AQ15: Incremental Learning of Attribute-Based Descriptions from Examples, the Method and User's Guide. Technical Report UIUCDCS-F-86-949. Department of Computer Science, University of Illinois at Urbana-Champaign.

Maloof, M. A., and Michalski, R. S. 1995. A Partial-Memory Incremental Learning Methodology and Its Application to Intrusion Detection. *Proceedings of the 7th IEEE International Conference on Tools with Artificial Intelligence*. November 5–8, Washington, DC (forthcoming).

Michalski, R. S. 1969. On the Quasi-Minimal Solution of the General Covering Problem. *Fifth International Symposium on Information Processing*, A3: 125–128.

Michalski, R. S. 1973. AQVAL/1 — Computer Implementation of a Variable-Valued Logic System VL_1 and Examples of Its Application to Pattern Recognition. *First International Joint Conference on Pattern Recognition*, 3–17.

Michalski, R. S. 1983. A Theory and Methodology of Inductive Learning. In Michalski, R. S.; Carbonell, J. G.; and Mitchell, T. M., eds., *Machine Learning: An Artificial Intelligence Approach*, Vol. 1, 83–134. Palo Alto, CA: Tioga Publishing

Michalski, R. S. 1985. Knowledge Repair Mechanisms: Evolution vs. Revolution. In Proceedings of the Third International Machine Learning Workshop, 116–119.

Mitchell, T.; Caruana, R.; Freitag, D.; McDermott, J.; and Zabowski, D. 1994. Experience with a Learning Personal Assistant. *Communications of the ACM* 37(7): 81–91.

Reinke, R. E., and Michalski, R. S. 1988. Incremental Learning of Concept Descriptions: A Method and Experimental Results. In Hayes, J. E.; Michie, D.; and Richards, J., eds., *Machine Intelligence 11*, 263–288. Oxford: Clarendon Press.

Schlimmer, J. C. (1987) Incremental Adjustment of Representations for Learning. In Proceedings of the Fourth International Workshop on Machine Learning, 79–90.