

The Logic of Learning

Pei Wang

Research Division, Intelligenesis Corporation

and

Center for Research on Concepts and Cognition, Indiana University

<http://www.cogsci.indiana.edu/farg/pwang.html>

Abstract

Most of the previous machine learning study is focused on algorithm-based learning. Another approach, reasoning-based learning, bring up many new research problems. Such a system, NARS, is briefly introduced. Finally, the difference between different types of learning is discussed.

Introduction

A machine learning system is often described as a “learning algorithm”, which takes raw data and background knowledge as input, and produces some output, usually a representation of a concept that was learned from the given data.

An “algorithm” is a computational process that, for the same input, always follows the same path, produces the same output at the end, and takes a constant amount of computational resources, namely (processor) time and (memory) space.

This paper is going to introduce a learning system, which does not fit the above description, therefore is not a “learning algorithm”. After a brief description of the major components of the system, it is compared with other learning approaches, and future research issues are discussed.

NARS, a Reasoning System that Learns from Experience

NARS (Non-Axiomatic Reasoning System) is an intelligent reasoning system. It answers questions according to the knowledge originally provided by its user. What makes it different from conventional reasoning systems is its ability to learn from its experience and to work with insufficient knowledge and resources.

Concretely, it means that the system should be open to new knowledge and questions in real time, and answer questions according to its available knowledge when the knowledge and resources are insufficient to provide a perfect answer.

A detailed description of NARS is in (Wang 1995). The current version, NARS 4.1, is a Java applet, which,

Copyright © 2000, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

and related publications, are available at the author’s web page. Limited by (conference) time and (publication) space, many issues can only be briefly mentioned in this paper. Interested (or confused) readers are strongly encouraged to visit the web page. Also, a demonstration of NARS 4.1 has been accepted by The Exhibit Program of AAAI-2000.

Knowledge Representation

NARS does not use first-order predicate logic. Instead, each piece of knowledge in NARS, called “judgment”, has the form “ $S \ r \ P \ < f, c \ >$ ”. Here S is the *subject term* of the judgment, and P is the *predicate term*. In the simplest situation, both of them are words. r is an *inheritance relation*. For the current discussion, three types of inheritance relation are involved:

- “ $S \subset P$ ” means that “ S is a special type of P ”;
- “ $S \in P$ ” means that “ S is an instance of P ”;
- “ $S = P$ ” means that “ S and P are similar to each other”.

The “ $\langle f, c \rangle$ ” is the truth value of the judgment, where f is the *frequency*, a real number in $[0, 1]$, indicating the ratio of positive evidence among all evidence of the relation, and c is the *confidence*, a real number in $(0, 1)$, indicating the amount of evidence the system has on the relation. In this way the truth value measures the relationship from a judgment to the system’s experience, not to an “outside world” or model.

Each question that can be asked to the system has the form $S \ r \ P$. A question looks like a judgment without a truth value, and where S or P (but not both) can be a special symbol “?”. A question without “?” is like a “yes/no” question — the system is asked to evaluate the truth value of the given relation. A question with “?” is like a “what” question — the system is asked to find a term that have more positive evidence and less negative evidence for the given relation.

Since the confidence of a judgment cannot reach 1.0, no judgment is absolutely certain. Instead, the system needs to compare among a set of candidates to decide a “best answer”, which may be overturned by new knowledge or further consideration.

Inference Rules

Each of the following basic rules in NARS takes two judgments as premises, and derive a new judgment as conclusion.

Revision $S \subset P \langle f_1, c_1 \rangle$ $S \subset P \langle f_2, c_2 \rangle$	Deduction $S \subset M \langle f_1, c_1 \rangle$ $M \subset P \langle f_2, c_2 \rangle$
$S \subset P \langle f, c \rangle$	$S \subset P \langle f, c \rangle$
Abduction $S \subset M \langle f_1, c_1 \rangle$ $P \subset M \langle f_2, c_2 \rangle$	Induction $M \subset S \langle f_1, c_1 \rangle$ $M \subset P \langle f_2, c_2 \rangle$
$S \subset P \langle f, c \rangle$	$S \subset P \langle f, c \rangle$
Analogy $S \subset M \langle f_1, c_1 \rangle$ $M = P \langle f_2, c_2 \rangle$	Comparison $S \subset M \langle f_1, c_1 \rangle$ $P \subset M \langle f_2, c_2 \rangle$
$S \subset P \langle f, c \rangle$	$S = P \langle f, c \rangle$

Since by definition $S \in P$ is identical to $\{S\} \subset P$, rules on the “ \in ” relation can be derived from those on the “ \subset ” relation.

Each rule has a truth value function that calculates the strength and confidence of the conclusion ($\langle f, c \rangle$) from those of the premises ($\langle f_1, c_1 \rangle$ and $\langle f_2, c_2 \rangle$). Different rules use different functions. The rules in NARS are truth-functional, because NARS does not use model-theoretic semantics, but rather use an “experience-grounded semantics”, so that the truth value of a conclusion is completely determined by the premises that derive it.

According to how the confidence c is calculated, the above rules can be put into three groups:

1. In Deduction and Analogy, if the premises have high confidence, so does the conclusion.
2. In Abduction, Induction, and Comparison, the confidence of the conclusion is always much lower than that of the premises.
3. Revision is the only rule where the confidence of the conclusion is higher than that of the promises, because this rule merges the evidence of the premises into that of the conclusion.

Besides these basic rules, NARS 4.1 also has compound-term composition and decomposition rules, such as “ $S \subset (P1 \wedge P2)$ ” if and only if “ $S \subset P1$ and $S \subset P2$ ”. Another type of rule is backward inference rule that derives a new question from a question and a judgment, such as from available knowledge “ $S \subset M$ ” and question “ $? \in M$ ” to derive a new question “ $? \in S$ ”, whose answer and the knowledge can derive an answer to the original question. This kind of rule allows the system to work in a goal-directed manner.

Control Mechanism

Assuming real-time input, NARS cannot work on a single task at a time, but must allow multiple tasks to

be under processing at the same time. Because of the assumption of insufficient knowledge and resources, it cannot assume that all tasks will be processed to their “logical end”, or to be solved by considering all relevant knowledge in the system.

Instead, the system processes multiple inference tasks by time-sharing. Each task is given a priority value, which indicates the frequency for it to be processed. After a task is selected for processing, a piece of knowledge is also selected according to a priority distribution. The combination of the task and the knowledge decides which inference rule can be applied, using the two as premises. The derived task and knowledge are put back into the task pool and knowledge base, and the priority values of the involved task and knowledge are adjusted, according to the feedback obtained in this inference step.

When an answer is found for a user question, it is reported, then the system continues to look for a better one, if the task still have a high enough priority.

Consequently, the processing of a user question no longer follows a predetermined algorithm, because it consists of a sequence of inference steps, and the combination depends on the constantly changing structure of the knowledge base and task pool. Also, when an answer is provided to the user, it is hard to tell whether it is the final answer, because it depends on future events. If the same question is asked to the system at different contexts, the answers may be different, so are the processing path and the time-space used on it.

The Demonstration

NARS has been implemented several times. The current version, 4.1, is a Java applet which is available at the author’s web page for on-line demo and download. Related documentation is available there, too.

The user interface of NARS 4.1 allows the user to provide knowledge and questions to the system in a text field. The system will return answers to the questions in another window. Since the timing of input influences the system’s processing, the user can also specify the number of inference steps allowed between input events.

The NARS 4.1 demo has a set of examples attached, and each of which shows a basic function or property of the system. The examples include: input and output, context sensitivity, deduction, induction, abduction, mixed inference, confidence processing, backward inference, contradiction handling, similarity evaluation, compound term formation, Hempel’s paradox, relation operators, and fuzzy concept formation.

In the on-line documentation, each example comes with a simple explanation about the system’s processing and the result, as well as links to related publications.

All of these examples can be given to the system by copy/paste. When a user becomes familiar enough to the system, he or she can test it with whatever example, as long as it can be put into the interface language of NARS.

Discussion

Learning in NARS

As described above, there are several types of learning going on in NARS:

- The inference rules generate new knowledge in each step, which is added to the knowledge base of the system.
- If a new conclusion has the same content as an existing piece of knowledge, the revision rule will merge the two, therefore changing the belief of the system according to new evidence.
- According to experience-grounded semantics, the meaning of a term in NARS is determined by the judgments in which the term appears. As new knowledge is generated and useless old knowledge forgot, the system learns the meaning of the terms according to its experience with them.
- The compound-term composition rules generate new terms from time to time. At the beginning, their meaning is determined by the meaning of their components. However, as the system gets more direct experience on them, they gradually become independent, and are treated for their own sake.
- By adjusting the priority distributions among terms, tasks, and knowledge, as well as by deleting useless ones, the system also learns what is important and relevant and so should be considered first when processing time is insufficient to consider everything.

Comparison of the two types of learning

To build a learning system in this way is very different from just building an algorithm which takes certain input and produces the desired output. The “logical approach” has several advantages over the traditional “algorithmic approach”:

- By working in real time, it allows different response-time requirements to be attached to a question. One question may need a quick answer, while another question may prefer a more carefully considered answer.
- New knowledge can be added from time to time, as well as by the request of the system. The system revises its beliefs incrementally, rather than restarts whenever new knowledge arrives.
- When it is impossible to consider all relevant knowledge, the system can make a rational selection according to experience and context.
- The selection of inference rules is data-driven, so neither the designer nor the user needs to specify how to answer a concrete question in advance.
- The learning process is integrated with reasoning, categorization, and problem solving. Actually in NARS they are different names of the same process.

- It is more similar to the learning process of human beings — we seldom learn new ideas by following a predetermined algorithm.

Of course, it does not mean that the logical approach is always better. Actually, whenever a learning algorithm is available and affordable, it usually gives a more reliable and efficient solution than NARS. On the other hand, something like NARS should be used when such an algorithm is not available (due to insufficient knowledge) or not affordable (due to insufficient computational resources).

NARS and ITL

In the current machine learning study, the closest work to NARS is the Inferential Theory of Learning (ITL) (Michalski 1994). Both NARS and ITL are inferential system that carry out multi-strategy learning, and they also share many theoretical and technical assumptions about machine learning, such as to understand learning as “a goal-guided process of modifying the learner’s knowledge by exploring the learner’s experience” (Michalski 1994).

These two approaches have the similar major components, but the technical decisions on each of them are quite different:

Knowledge representation: While ITL uses predicate logic for knowledge representation, the formal language used by NARS belongs to the “term logic” tradition, which includes Aristotle’s syllogism.

Semantics: In ITL, “truth” is defined according to model-theoretic semantics, and is different from the subjective “certainty” measurement. In NARS, truth value includes a frequency factor and a confidence factor, and is a measurement of the available evidence for the given proposition.

Inference rules: The two systems have different rule sets. Though both include deduction, induction, abduction, and analogy, the exact definitions are not the same. Because of the experience-grounded semantics, inference rules are “truth functional” in NARS, which is not the case in ITL.

Knowledge organization: In NARS, priority distributions are maintained among tasks and pieces of knowledge, so that tasks are processed at different rates and pieces of knowledge have different probabilities of being used. By adjusting the priority distributions, the system learns control and context information. I have not found a corresponding mechanism in ITL.

Control mechanism: ITL characterizes a learning process as a goal-guided search through a knowledge space. NARS processes its tasks by interacting them with available knowledge at different rates to find matching answers and to derive new knowledge and tasks. This process does not follow a predetermined algorithm.

It will be interesting to compare NARS and ITL in detail, so as to get some conclusions about the logical/inferential approach in general, as well as the differences (and their implications) of the two approaches.

Open issues

Since NARS is still under development, and the logical approach of learning has not been explored much by the machine learning community, there are still more open issues than sure conclusions.

The current version of NARS needs to be extended further to including higher-order inference (to infer according to the implication and equivalence relations among statements) and procedural knowledge (to infer about events and actions), among other things. The control mechanism of NARS also needs to be refined and extended to handle complex problems.

To compare different types of machine learning approaches in a detailed level will bring up many new research issues. In particular the relationship between reasoning and learning has not received enough attention in the machine learning community. The purpose of the current paper is not to present concrete conclusions, but to raise people's interest in inference based learning.

References

Dietterich, T. G. 1997. Machine Learning Research: Four Current Directions. *AI Magazine* 18(4):97-136.

Kubat, M., Bratko, I. and Michalski, R.S. 1998. A Review of Machine Learning Methods, in Michalski, R.S., Bratko, I. and Kubat, M. (eds.) *Machine Learning and Data Mining: Methods and Applications*, London: John Wiley & Sons, 3-69.

Michalski, R. S. 1994. Inferential Theory of Learning: Developing Foundations for Multistrategy Learning, in Michalski, R.S. and Tecuci, G. (eds.) *Machine Learning: A Multistrategy Approach*, Vol. IV, Morgan Kaufmann, San Mateo, CA.

Wang, P. 1995. Non-Axiomatic Reasoning System: Exploring the essence of intelligence. Ph.D. diss., Dept. of Computer Science and Program of Cognitive Science, Indiana Univ.