Application of Artificial Intelligence to Adaptive Instruction - Combining the Concepts

Jan Joris Roessingh, Gerald Poppinga, Joost van Oijen(✉), and Armon Toubman

Netherlands Aerospace Center NLR,
Anthony Fokkerweg 2, 1059 CM Amsterdam, The Netherlands
{Jan.Joris.Roessingh, Gerald.Poppinga, Joost.van.Oijen, Armon.Toubman}@nlr.nl

Abstract. In recent years, instructional systems for individuals and teams, including virtual environments, serious games, simulator-based training and on-the-job/live training, have been supplemented by Adaptive Instructional Systems (AISs). Artificial Intelligence (AI) techniques and Machine Learning (ML) techniques have been proposed, and are increasingly used, for a number of functions of AISs. This paper aims to combine, on the one hand, the concepts of AI and ML, and, on the other hand, adaptive instruction. The emphasis is put on simulator-based training in a professional context, predominantly skill learning by practicing tasks in simulated environments, either as an individual student or as part of a team. The major goals of this paper are: (1) to provide a basic description of available ML techniques, (2) to sketch the potential use of machine learning techniques in adaptive instruction, and (3) to provide examples of applications from the literature. This paper neither introduces a new AI approach to adaptive instruction, nor does it extensively review the literature of such approaches.

Keywords: Intelligent Tutoring Systems · Adaptive instruction · Artificial Intelligence

1 Introduction

In recent years, instructional systems for individuals and teams, including virtual environments, serious games, simulator-based training and on-the-job/live training, have been supplemented by Adaptive Instructional Systems (AISs). This is a general term for intelligent, computer-based tools used for education, instruction and training. AISs guide learning experiences by tailoring instruction and recommendations based on the goals, needs and preferences of each learner (or team) in the context of domain learning objectives [1]. Examples of AISs are Intelligent Tutoring Systems (ITSs), intelligent mentors and personal assistants for learning. Such technology has been integrated in computerized instructional media, including training systems that are embedded in operational systems (so called embedded training systems). Artificial Intelligence (AI) techniques and Machine Learning (ML) techniques have been

R. A. Sottilare and J. Schwarz (Eds.): HCII 2019, LNCS 11597, pp. 542–556, 2019.
https://doi.org/10.1007/978-3-030-22341-0_43

proposed, and are increasingly used, for a number of functions of AISs (see [2–4]). These functions include expert domain knowledge representation, predicting the behavior, state and final performance of students, preventing drop-out of students, generating the behavior of non-player characters (NPCs) in training scenarios, and, more generally, the generation of suitable training scenarios and orchestrating pedagogical interventions. Examples of use of AI in scenario management and serious gaming packages for the military are provided in [5]. More recently, [6] provided several examples of the use of machine learning techniques for behavior generation in simulator-based training. The current paper aims to combine, on the one hand, the concepts of AI, more specifically ML techniques, and, on the other hand, adaptive instruction. The emphasis is on simulator-based training in a professional context, predominantly skill learning by practicing tasks in simulated environments, either as an individual student or as part of a team.

The purpose of this paper is to introduce the HCII AIS conference session on application of AI to adaptive instruction. The major goals of this paper are: (1) to provide a basic description of available ML techniques, (2) to sketch the potential use of machine learning techniques in adaptive instruction, and (3) to provide examples of applications from the literature. This paper neither introduces a new AI approach to adaptive instruction, nor does it extensively review the literature of such approaches.

2 Machine Learning

AI is applied when the machine mimics “cognitive” functions that humans associate with the human mind, such as “learning” and “problem solving” [7]. The field of ML grew out the broader study of AI over the past several decades. As a research field, ML studies computer algorithms that improve automatically through experience [8]. In a data processing sense, ML, which is also known as ‘predictive analytics’, is a set of techniques with several prominent applications, such as detection, clustering, classification, prediction, dimension reduction, decision support, simulation and data generation, in particular through non-pre-programmed methods. Traditional AI methods for data processing and prediction have relied on explicitly programmed systems, often captured as a set of “if-then” rules of which a predictable subset would be activated under specified conditions. However, such expert systems only performed as well as their original programming, and were often unable to adapt to unexpected data sets, thus lacking robustness.

ML, however, attempts to apply the concepts of AI in a way that allows a computer program to improve its performance over time, without being explicitly programmed. ML approaches generally have the implicit benefit of producing a reasonable satisfactory solution in the face of new, unknown, observations. In such cases, traditional expert systems might get stuck. The downside of producing such solutions, however, is that it can be unclear to humans how ML algorithms produce these solutions, which doesn’t help in interpreting, since the relationship between input and output may be difficult to infer and cannot be explained.

From a historical perspective, the application of ML to adaptive instruction, in the current day appreciation, i.e. beyond the Skinner teaching machine [9, 10], started around 1970 with Carbonell [11, 12], who developed a prototype ITS. This system, called SCHOLAR, was "[..] information structure oriented, based on the utilization of a symbolic information network of facts, concepts, and procedures." A semantic network is a set of concepts, like 'electrons', 'neutrons', 'protons' and 'atoms', and relations among those concepts, such as, 'the nucleus of an atom consists of neutrons and protons', and 'electrons orbit the nucleus of an atom'. SCHOLAR's semantic network had the capability to learn such patterns from facts ("SCHOLAR learns what is told.", [12, p. 52]), but main concern in the development was how to use and represent knowledge in the semantic network. Main feature of SCHOLAR was its ability to maintain a 'mixed-initiative' dialogue with the student, with questions asked by either SCHOLAR or the student and answered by the other. In the decade thereafter, adaptive instruction became a research-intensive subfield of ML, with many practitioners from the cognitive- and computer sciences, as apparent from, for example, [13]. In summary, whereas traditional AI tried to create self-contained expert systems with steady output, current day ML aims to create a system that can, through repeated exposure to data, learn for itself and therewith able to adapt to novel data.

2.1 Machine Learning Approaches

Domingos [14] subdivides ML methods as follows:

- Evolutionary, ML is considered as the natural selection through the mating of computer programs, with genetic programs as the representational form of this class of methods. [15] provides an example of using this type of method in adaptive instruction.
- Connectionism, ML is considered as adjusting weights in Artificial Neural Networks (ANNs), the ANNs being the representational form of this class of methods.
- Symbolic reasoning, ML is considered as logical deduction from symbolic information ('symbol manipulation'). Logic symbolic systems, in the form of sets of rules or decision trees, are the representational form of this class of methods. The aforementioned SCHOLAR ITS [11, 12] would be an example of application of this method.
- Bayesian probabilistic reasoning, ML of probabilities (of events or the truth of propositions) is based on inference, i.e. the propagation of probabilities through a graphical network. The latter network is the representational form for this class of methods, in which the nodes describe events, and the arcs the relationships between events.
- Analogy-based learning, ML is based on determining similarity between data points. Representational forms for this class of methods are the nearest neighbor algorithm and support vector machines. These methods are able to find a decision rule for binary classification (decide on one of two outcomes), based on several predictor variables.

Although each of these approaches has their preferred set of methods, tools and techniques, they are not mutually exclusive and allow for hybrid approaches.

2.2 Three Types of Machine Learning from an Input-Output Perspective

ML can also be looked at from a perspective of how the ML algorithm operates. What is needed at the input? How is the output structured? Then, basically three types of machine learning can be distinguished: (1) supervised learning, (2) unsupervised learning, and (3) reinforcement learning.

Supervised Learning. This type of ML encompasses the techniques that learn from examples. For supervised learning techniques, the data (examples, observations) must be labeled, i.e., annotated by a human expert or some other source. In other words, a human or another entity supervises the learning process of the machine. The algorithm forms a model about relations (features) between the data and the labels. For example, a supervised learning algorithm learns to classify correct and incorrect student responses after it has been presented with a large number of such responses, labeled by an expert instructor.

Unsupervised Learning. This type of ML includes techniques that automatically build a model of the data that they are presented with. For unsupervised learning techniques the input data is unlabeled, i.e., not annotated by an expert, in other words, unsupervised. The algorithm forms a model of the structures or patterns that are present in the data, without any explicit hints as to what those structures are. An unsupervised ML algorithm may discover patterns in student behavior that are not directly apparent to human instructors, for example, that good student performance on teamwork aspects in a simulator-based training is strongly associated with a certain cluster of measures.

Reinforcement Learning. This type of ML can be characterized with ‘learning by doing’. Reinforcement learning is often associated with a learning agent, i.e. a robot or software agent, that learns through interaction with some system or environment. Through its behavioral actions it interacts with the environment and therewith creates examples of this behavior. The agent receives a reward for its actions when it achieves some goal, such as solving a maze, completing a game, making a profitable deal, or neutralizing an opponent. The received reward serves as a kind of label with the example, i.e. the series of actions undertaken by the agent to reach its goal. The agent therewith learns to gradually improve the sequence of actions (‘the policy’) to achieve its goal.

As with the five ML approaches discussed in the previous section, hybrids of different types of ML are often successful. For example, many of the successes with deep (reinforcement) learning (e.g. [16, 17]) are based on algorithms that combine supervised, unsupervised and reinforcement learning in a connectionist approach.

On-Line Learning and Off-Line Learning. Another useful distinction is between on-line learning and off-line learning. When a ML application receives its training, the algorithm will most often set some balance between exploration (actions that create or include novel examples, but with uncertain reward) and exploitation (actions that are ‘known’ by the algorithm to lead to the desired reward). Both are essential for learning. However, in practical applications, exploration by the algorithm may be undesirable, for example, because it may lead to emergent or unstable behavior of the algorithm, which would disturb the tutoring of an ITS. Therefore, with off-line learning, the algorithm will be trained before actual application, and will not be allowed to continue learning during application. With on-line learning, learning (and therewith some degree of exploration) continues during practical application.

3 What Is Adaptive Instruction?

In an educational context, Gaines [18] describes adaptive instruction as sketched in Fig. 1. In his view, an adaptive instructional system has three elements: (1) the evaluation of learning outcome, (2) a dynamic model of the learning process that is implemented in the so-called adaptive logic, and (3) an adaptive variable that changes the training task or the environment. The learning process is the result of interactions between a student (or a team) and a task. The control model is a model of the learning process. The parameters of this model are not fixed but will be selected on the basis of the observed learning process. Examples of adaptive instruction are:

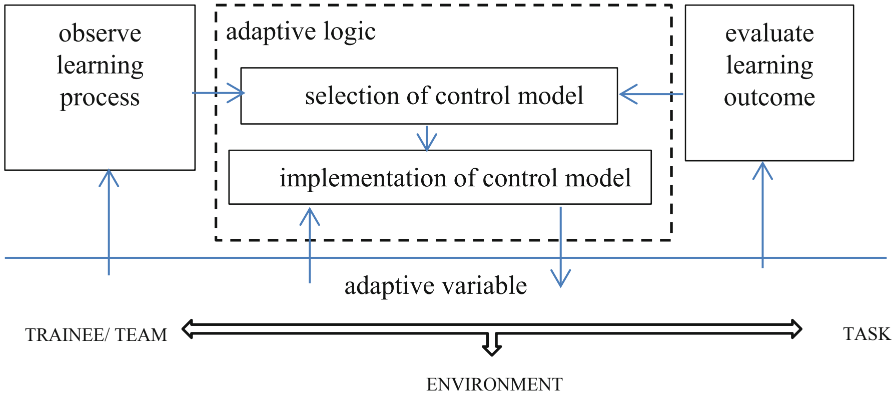

Fig. 1. Concept for adaptive instruction (based on [18]). The modules above the horizontal line establish the Artificial Intelligence. Interaction between trainee/team, task and environment is denoted by the black arrow. The adaptive variable may modify any of these.

- The control we take over our own learning process, based on an implicit model of that learning process;
- The control that an instructor takes over the learning process of a student, based on an implicit model of the learning process of the student;
- The control of a learning process of a student by an algorithm that explicitly models a learning process of the student.

The first two of these examples are examples of adaptive instruction based on human intelligence, while the latter example may be based on an (AI-) algorithm, possibly a ML algorithm. Hence, in AI-based adaptive instruction, the educational experience is tailored by an AI-enabled tutor. More generally, the goal of adaptive instruction is to optimize learner outcomes [19]. Learner outcomes that can be optimized are, for example, knowledge and skill acquisition, performance, enhanced retention, accelerated learning, or transfer-of-training between different instructional settings or work environments. A range of technologies may fall under the heading of AISs, including personal assistants for learning and recommender systems. For the current purposes, we consider the ITS as the most comprehensive technology under this heading, in a sense that it is not just an add-on, but creates the complete setting for tutoring: domain expertise, a tutor, and communication or other interactions with the student.

According to [20], an ITS is an educational support system (a kind of virtual tutor), used to help learners in their tasks and to provide them with specific and adapted learning contents. Nwana [21] emphasizes that ITSs are designed to incorporate techniques from the AI community in order to provide tutors which know *what* they teach, *who* they teach, and *how* to teach it. Hence, less comprehensive AIS technologies may not be capable of knowing the full scope of teaching (*what*? *who*? *how*?). An ITS would provide instant and personalized instruction or feedback to students, usually without needing involvement of a human tutor, with the purpose of enabling learning in a meaningful and effective manner. Many situations, such as on-the-job, in the classroom or in remote locations, lack sufficient availability of one-to-one instruction, the latter being more effective than one-to-many instruction. In such situations, an ITS is capable of mimicking a personal tutor (one instructor per student or per team). There are many examples of ITSs being used in such situations, including aerospace, military, health care and industry, and where their capabilities and limitations (see e.g. [22]) have been demonstrated. Machine learning techniques may be helpful to further enrich these capabilities and mitigate the limitations. In this paper, we will therefore consider the application of machine learning techniques to the various components of the ITS.

4 Modules of an ITS

The general concept of an ITS [21, 23, 24] is based on four modules (see Fig. 2). Arrows denote the exchange of information. For example, the domain expert module provides performance standards to the student module. The tutoring module receives progress information (from the student module) on a learning objective (selected by the tutoring module) and plans the next exercise. The exercise will be made available to the student via the user interface module. The student provides his/her response back to the user interface module, etc. The four modules are now briefly discussed.

4.1 The Domain Expert Module

The first module, the domain expert module, contains the concepts, rules, and problem-solving strategies for the domain to be learned. Its main function is to provide the standard for evaluating the student's response and therewith providing the ability to assess the student's or team's overall progress.

Expert knowledge must not only include shallow knowledge (e.g. the categories and explanations of various concepts that the student has to acquire), but also the

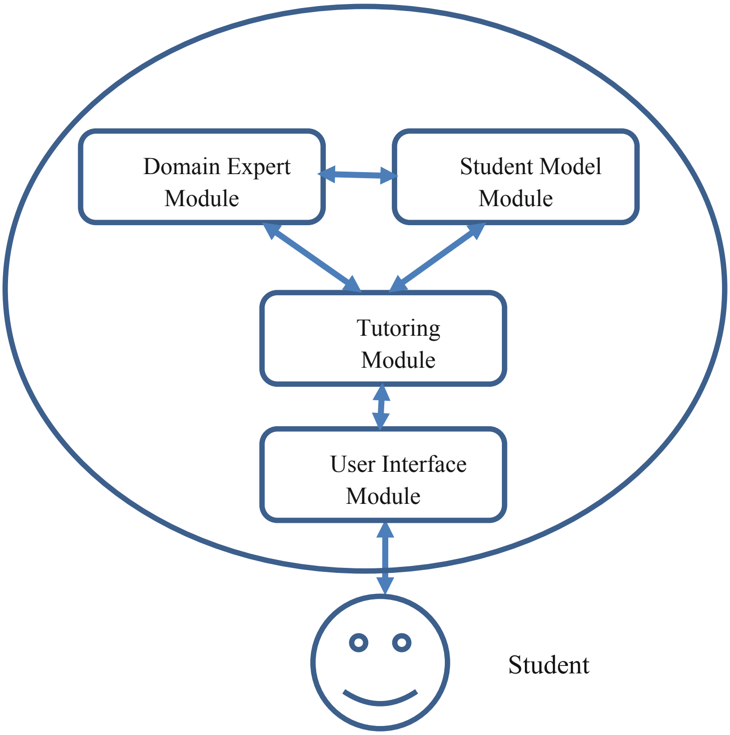

Fig. 2. General concept of an ITS and its relation to a student (from [21])

representational ability that has been acknowledged to be an essential part of expertise. Expert knowledge can be represented in various ways, including network presentations (e.g. belief networks), (rule-based) production systems, behavior trees, hierarchical finite state machines, or as a set of constraints, which can be used to analyze students' solutions in order to provide feedback on errors.

Knowledge elicitation and codification of this knowledge can be very time-consuming, especially for a complex domain with an enormous amount of knowledge and interrelationships of that knowledge. Thus, investigating how to encode knowledge and how to represent it in an ITS remains the central issue of creating an expert knowledge module [21]. Please note that this bears similarity with issues in knowledge representation and explainability in ML and AI in general.

The expert module of an ITS should also be viewed in the context of simulator-based training or in a gaming/virtual environment. Is such an environment, the student has to learn by doing how to perform a given task, e.g. with the goal to defeat an enemy or to troubleshoot and resolve a malfunction in a piece of equipment, etc. In such context, not only the explicit knowledge, but also the simulation itself is part of the domain expert module, with its built-in concepts, behavior of NPCs, rules, constraints and score keeping for indicating task performance.

4.2 The Student Model Module

The student model module refers to the dynamic representation of the evolving knowledge and skills of the student, as would become apparent from a ‘learning curve’ for this student. Important functions of the student model are: (1) to evaluate a student’s or a team’s competency with the tasks to be mastered against the standard established in the domain expert module, and (2) to evaluate how competency evolves with further exposure to the current state of the learning environment. The results of these evaluations feed into the tutoring module (to be discussed in the following subsection), that will decide on pedagogical adaptations of the learning environment. The student model module thus acts as a source of information about the student. Such knowledge is vital for the tutoring module of the ITS, as no intelligent tutoring can take place without such understanding of the student.

The student model should include those aspects (variables) of the student’s behavior and knowledge that have an assumed effect on his/her performance and learning. Constructing a valid model is non-trivial. Human tutors would normally combine data from a variety of sources, possibly using bodily posture, gestures, voice effects and facial expressions. They may also be able to detect aspects such as boredom or motivation which are crucial in learning. The evolution of these cognitive and affective states must then be traced as the learning process advances. However, in the absence of a human tutor, the student’s cognitive and affective states must be inferred from the student input received by the ITS, via a keyboard, and/or other input devices or sensors.

Traditionally in ITSs, a student model could often be created from the representation of the target knowledge in the expert knowledge module. Accordingly, the student model can include a clear evaluation of the mastery of each unit of knowledge in the expert module. This allows the student’s state of knowledge to be compared with the expert knowledge module, and instruction would then be biased towards portions of the model shown to be weak. This form of student modelling is referred to as ‘overlay’ modelling [25], because the student’s state of knowledge is viewed as a subset of the expert domain knowledge. Thus, in this form, the student model can be thought of as an overlay on the domain model. As the student progresses though the training tasks and the student model starts to deviate from the domain expert model, this is flagged to the tutoring module.

4.3 The Tutoring Module

The tutoring module is the part of the ITS that plans and regulates teaching activities (see Fig. 2) via the user interface module. In other architectures, this module is referred to as the teaching strategy or the pedagogic module.

It plans the teaching activities on the basis of information from the student model module about the student’s learning progress relative to the objectives defined in the domain expert module. The tutoring module thus decides on activities to achieve learning objectives: hints to overcome impasses in performance, advice, support, explanations and different practice tasks (see e.g. [26]). Such decisions or suggestions are based on the instructional strategy of the tutoring module, the evolution of the student's competencies and possibly the student's profile (see e.g. [20]).

The sequence and way in which activities take place can lead to distinct learning outcomes. Tightly orchestrating the teaching activities might harm the student's explorative abilities. Sometimes, it may be more effective to let the student struggle for some time before interrupting. However, the student should not lose his motivation when he or she gets stuck during struggling.

In traditional implementations of ITSs, for example an application to learn to resolve algebra problems, the student may request guidance on what to do next at any point in a problem-solving process (e.g. [27]). This guidance is then based on the comparison between the student's state of knowledge and the expert knowledge. The tutoring module diagnoses that the student has turned away from the rules of the expert model and provides feedback accordingly. In a similar fashion, e.g. in an application that teaches the programming language Lisp [28], every time a student successfully applies a rule (from the domain expert module) to a problem, the student model module increases a probability estimate that the student has learned the rule. The tutoring module keeps on teaching students on problems that require effective application of a rule until the probability that the rule has been learned exceeds a certain criterion. The tutoring in existing ITSs can be ordered along a range with an increasing flexibility of control. At the low end, there are systems that monitor every response of the student diligently, adjusting the tutoring activities to the student's responses but never resigning control. At the high end, there are guided discovery learning systems where the student has maximum control over the activity, and the only way the system can direct the course of action is by modifying the environment. Somewhere halfway this range are more versatile tutors, where the control is shared by the student and the ITS as they exchange information. The presence of this variety in tutoring styles underlines that variation in flexibility is required for different applications and possibly at different stages of the student's learning process. Such tutoring requirements are still challenging to formulate and to embody in an ITS. Nevertheless, some progress has been made, and machine learning techniques will certainly help to create the potential to adapt and improve strategies over time (as in the case of self-improving tutors), and for the same strategies to be used for other domains.

4.4 The User Interface Module

The user interface module regulates a dialogue between the student and the tutoring module, as depicted in Fig. 2. It translates between the tutoring module's internal representation of the teaching activities and the behavior of the student in a communication form that is on the one hand comprehensible to the student and, can, on the other hand, be processed by intelligent tutor.

The consideration of the user interface module as a distinct part of the ITS architecture should lead to explicitly addressing of user-interface design and usability issues during ITS development [29]. Challenges that relate to the user interface are: ease of use; natural interaction dialogues; a dialogue that is task-oriented and adaptive; effective screen design; and supporting a variety of interaction styles and/or learning styles. No matter how 'intelligent' the internal system is, if these challenges have not been suitably addressed, the ITS is unlikely to yield positive transfer of learning and become acceptable for the student.

Progress in user interface design is progressively delivering superior tools whose interactive capabilities strongly influences ITS design. ITSs provide user interfaces which, for the input, range from the use of fixed menus with multiple-choice answers to the use of natural language, gestures, hand-, finger- and eye-movements, use of 3D-pointing devices and a variety of physiological sensors/measurements. For the output, they range from the mere display of pre-stored texts, to the use of computer-generated speech and multi-modal virtual reality displays. Within these two ends of the range, the designers are, in principle, flexible in their choice. Much more experimental research into the use of such user interfaces is still required.

5 Applying ML to Adaptive Instruction

Both adaptive instruction and ML are terms for a broad set of technologies and vast fields of research. In the aforementioned sections we have broken down a commonly used ITS concept in its modules: the domain expertise, the student model, the tutor, and the user interface. Given a specific application, it may be evaluated whether required functionality of each module can benefit from ML. Whether or not it is worthwhile to apply a ML technique in the implementation of such functionality must be evaluated at a case-by-case basis by the ITS designers. ML is notably strong in tasks that require detection, clustering, classification, prediction, dimension reduction, decision support, simulation and data generation. There are many software tools and libraries that provide ML solutions in support of these tasks. It may well be possible to design ITSs that do not perform these tasks, and if so, they might as well be done manually, or implemented with techniques that do not fall under the heading of ML techniques as discussed in Sect. 2. In the following we provide examples, per module, of where ML could be a feasible technique to fulfill a functional requirement.

5.1 Applying ML to the Domain Expert Module

The main function of the expert module is to provide expert knowledge or expert behavior that can serve as the basis for evaluating the student's response. In tactical training, particularly in the military, trainees often have to respond to other parties that can be friendly, cooperative, hostile or neutral. In virtual games and simulator-based environments, these parties take the form of NPCs. The behavior of NPCs is part of the domain expertise. Specifying NPC behavior through manual programming of an ITS, can be an expensive, time consuming and tedious job, requiring specialized personnel. Several examples of behavior generation for NPCs using ML are provided in [6]. Different ML techniques have been used in different applications to overcome the knowledge elicitation challenge. A rule-based reinforcement learning technique called Dynamic Scripting [30] has been applied to generate behavior of opponents in an air combat training system (see [31]). A different ML technique, so-called Data Driven Behavior Modeling (DDBM) has been applied (see [32]) to create NPCs in VBS3 (Virtual Battle Space, a game-based military simulation system). These NPCs learn bounding overwatch for dismounted infantry, which is a military tactical movement used to improve the security of units when they are moving towards a target.

5.2 Applying ML to the Student Model Module

One function of the student model is to evaluate how competency evolves with further exposure to the current state of the learning environment. In a study [39], an intelligent agent was developed with the aim to mimic student learning behavior. The agent managed to learn a complex game (the Space Fortress game) using reinforcement learning as an ML technique. Some learning characteristics, such as transfer-of-training between part-tasks, were comparable to that of human students. Hence the model may, in principle, be used to predict student learning characteristics as part of the student model.

5.3 Applying ML to the Tutoring Module

An important function of the tutoring module is to plan the teaching activities on the basis of information from the student model module. A genetic algorithm in combination with novelty search and combinatorial optimization, to automated scenario generation was used in [19]. As an example, the "clear rooms training task" [40] was used, in which a team of soldiers has to learn to clear rooms with various complexity factors. The series of generated scenarios has been made adaptive to a current competency measurement of the team. In a more general sense, self-improving tutors can be devised using ML to adapt the learning environment. Adaptive variables such as (1) error-sensitive feedback, (2) mastery learning, (3) adaptive spacing and repetition for drill and-practice items, (4) fading of worked examples for problem solving situations, or fading of demonstrations for behavioral tasks (such as in scenario-based simulations), and (5) metacognitive prompting, both domain relevant and domain independent, were suggested in [33]. For adaptive variables in flight simulations, [34] suggested aspects of the (simulated) environment such as illumination, sound-level, turbulence, g-forces, oxygen supply or manipulation of controls, displays and task load.

5.4 Applying ML to the User Interface Module

The user interface provides the translation between the tutoring module's plans and the behavior of the student in a form that is comprehensible to the student and can be processed by the intelligent tutor. Natural language processing (speech recognition, natural language understanding, natural language generation) is an aspect of the user interface module where ML could be an applicable. Moreover, Cha et al. [35] appreciate that each learner has different preferences and needs. Therefore, it is crucial to provide students with different learning styles with different learning environments which are suited to their preferences and provide a more efficient learning experience to them. Cha et al. report a study of an ITS where the learner's preferences are detected, and then the user interface is customized in an adaptive manner to accommodate the preferences. An ITS with a specific interface was created based on a learning-style model. Different student preferences became apparent through user interactions with the system. Using this interface and using different ML techniques (Decision Tree and Hidden Markov Model approaches), learning styles were diagnosed from behavioral patterns of the student interacting with the interface.

6 Discussion/Conclusion

ML can potentially be applied in adaptive instruction for any domain requiring trained operators and teams. Life-long learning, increasing demand for education and training, technological progress, and the scalability of adaptive instruction will supposedly contribute to its spread. However, new developments may spread slower than we expect. As an example, it was generally expected that Massive Open Online Education Courses (MOOCs) would disrupt existing models of higher education. However, recent research (e.g. [36]) into their success reveals that most students do not complete such courses. Dropout rates of MOOCs offered by Stanford, MIT and UC Berkley are as high as 80–95%. Given that the first prototypes of ITSs were already developed in the early seventies, and one may assume that lessons learned from adaptive instruction could be applied to MOOC design in order to increase their success. This suggests that further progress can be booked in this area.

In this paper, we present an architecture to discuss the application of ML techniques to adaptive instruction, particularly ITSs. In the sense of the model of Fig. 2, ML can potentially be applied in all four modules of the model. This is supported by concrete examples of the application of ML techniques in the previous sections.

The tutoring module decides how learner outcomes have an effect on the adaptive variable. Such decisions must be better tailored to the individual student or team than decisions built into non-adaptive computer-based instruction. The tutoring module takes into account the characteristics of the learning process of the individual or team. Changes in an adaptive variable can then be tailored to this learning process. This distinguishes adaptive from non-adaptive instructional strategies. It implies that adaptive instruction has added value over non-personalized instruction in applications where individuals or individual teams have sufficiently different learning processes and learner outcomes.

For the purpose of evaluation of learner outcomes, valid behavioral markers must be defined. In turn, these must be represented by signals that can be used in evaluation. For example, in a relatively straightforward task, such as a compensatory tracking task, the goal is to minimize the deviation, i.e. the difference between a manual output signal and a reference signal. A valid and reliable learner outcome may be found through averaging this deviation over a certain time period. For more complex real-life tasks, the determination of learner outcomes of interest, associated behavioral markers and processing of signals that represent these markers, are equally complex. ML techniques may be of help to find solutions in this context, too.

The application of ML may also have some potential disadvantages. Methods may be opaque in the sense that the relationship between input and output neither be inferred, nor explained. Some methods require massive numbers of data to converge to a solution, which numbers cannot always be made available in an educational context. Emergent, unstable or unexpected behavior of ML-enabled functions may be problematic for instructors, but possibly also for other purposes. Some methods are resource intensive and computationally heavy that may render them unsuitable for e.g. mobile platforms and real-time application. Also, in some settings, it may be desirable that a human instructor temporarily takes over control from the intelligent tutor (or the expert module). This may constrain the use of ML for specific purposes or applications. ML techniques, the behavior models they generate, and the tools with which they are controlled should facilitate such takeovers, and the behavior of the ITS should adapt gracefully.

References

1. Sottilare, R.A.: Applying adaptive instruction to enhance learning in non-adaptive virtual training environments. In: Bagnara, S., Tartaglia, R., Albolino, S., Alexander, T., Fujita, Y. (eds.) IEA 2018. AISC, vol. 822, pp. 155–162. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-96077-7_16
2. Hämäläinen, W., Vinni, M.: Comparison of machine learning methods for intelligent tutoring systems. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 525–534. Springer, Heidelberg (2006). https://doi.org/10.1007/11774303_52
3. Kotsiantis, S.B., Pierrakeas, C.J., Pintelas, P.E.: Preventing student dropout in distance learning using machine learning techniques. In: Palade, V., Howlett, R.J., Jain, L. (eds.) KES 2003. LNCS (LNAI), vol. 2774, pp. 267–274. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45226-3_37
4. Minaei-Bidgoli, B., Kashy, D.A., Kortemeyer, G., Punch, W.F.: Predicting student performance: an application of data mining methods with an educational web-based system. In: Frontiers in Education (FiE), p. T2A-13. IEEE (2003)
5. Abdellaoui, N., Taylor, A., Parkinson, G.: Comparative analysis of computer generated forces' artificial intelligence. Technical report, Defence Research and Development, Canada, Ottawa (Ontario) (2009)
6. Roessingh, J.J.M., et al.: Machine learning techniques for autonomous agents in military simulations - multum in parvo. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff Center, Banff, Canada (2017)
7. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education, Upper Saddle River (2003)
8. Mitchell, T.: The discipline of machine learning. Technical report CMUML-06-108. Carnegie Mellon University, Pittsburgh, PA 15213 (2006)
9. Skinner, B.F.: Teaching machines. Science **128**(3330), 969–977 (1958)
10. Skinner, B.F.: Review lecture - the technology of teaching. Proc. R. Soc. **162**, 427–443 (1965)
11. Carbonell, J.R.: AI in CAI: an artificial intelligence approach to computer-assisted instruction. IEEE Trans. Man-Mach. Syst. **II**, 190–202 (1970)
12. Carbonell, J.R.: Mixed-initiative man-computer instructional dialogues. Final Report, BBN Report No. 1971, Job No. 11399. Bolt Beranek and Newman, Inc., Cambridge, MA, USA (1970)
13. Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (eds.): Machine Learning: An Artificial Intelligence Approach. Springer, Heidelberg (1983). https://doi.org/10.1007/978-3-662-12405-5
14. Domingos, P.: The Master Algorithm: How the Quest for the Ultimate Learning Machine will Remake Our World. Basic Books, New York (2015)

15. Sottilare, R.: A hybrid machine learning approach to Automated Scenario Generation (ASG) to support adaptive instruction in virtual simulations and games. In: The International Defense & Homeland Security Simulation Workshop of the I3M Conference, Budapest, Hungary, September 2018
16. Hessel, M., et al.: Rainbow: combining improvements in deep reinforcement learning. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (2018)
17. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**, 529–533 (2015)
18. Gaines, B.R.: The learning of perceptual-motor skills by men and machines and its relationship to training. Instr. Sci. **1**(3), 263–312 (1972)
19. Sottilare, R.A., Burke, C.S., Salas, E., Sinatra, A.M., Johnston, J.H., Gilbert, S.B.: Designing adaptive instruction for teams: a meta-analysis. Int. J. Artif. Intell. Educ. **28**(2), 225–264 (2017). https://doi.org/10.1007/s40593-017-0146-z
20. Fonte, F.A.M., Burguillo, J.C., Nistal, M.L.: An intelligent tutoring module controlled by BDI agents for an e-learning platform. Expert Syst. Appl. **39**(8), 7546–7554 (2012)
21. Nwana, H.S.: Intelligent tutoring systems: an overview. Artif. Intell. Rev. **4**(4), 251–277 (1990)
22. Kulik, J.A., Fletcher, J.D.: Effectiveness of intelligent tutoring systems: a meta-analytic review. Rev. Educ. Res. **86**(1), 42–78 (2016)
23. Freedman, R.: A plan manager for mixed-initiative, multimodal dialogue. In: AAAI 1999 Workshop on Mixed-Initiative Intelligence (1999)
24. Nkambou, R., Bourdeau, J., Mizoguchi, R.: Advances in Intelligent Tutoring Systems. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14363-2
25. Goldstein, I.P.: The genetic graph: a representation for the evolution of procedural knowledge. Int. J. Man Mach. Stud. **11**(1), 51–77 (1979)
26. Self, J.A.: Student models: what use are they? In: Ercoli, P., Lewis, R. (eds.) Artificial Intelligence Tools in Education, pp. 73–86. North Holland, Amsterdam (1988)
27. Koedinger, K.R., Anderson, J.R., Hadley, W.H., Mark, M.A.: Intelligent tutoring goes to school in the big city. Int. J. Artif. Intell. Educ. (IJAIED) **8**, 30–43 (1997)
28. Corbett, A.T., Anderson, J.R.: Student modeling and mastery learning in a computer-based programming tutor. In: Frasson, C., Gauthier, G., McCalla, G.I. (eds.) ITS 1992. LNCS, vol. 608, pp. 413–420. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-55606-0_49
29. Padayachee, I.: Intelligent tutoring systems: architecture and characteristics. In: Proceedings of the 32nd Annual SACLA Conference, pp. 1–8 (2002)
30. Spronck, P., Ponsen, M., Sprinkhuizen-Kuyper, I., Postma, E.: Adaptive game AI with dynamic scripting. Mach. Learn. **63**(3), 217–248 (2006)
31. Toubman, A., Roessingh, J.J., Spronck, P., Plaat, A., Van Den Herik, J.: Rewarding air combat behavior in training simulations. In: IEEE International Conference on Systems, Man, and Cybernetics, Hong Kong, pp. 1397–1402 (2015)
32. Kamrani, F., Luotsinen, L.J., Løvlid, R.A.: Learning objective agent behavior using a data-driven modeling approach. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, pp. 002175–002181, October 2016
33. Durlach, P.J., Ray, J.M.: Designing adaptive instructional environments: in-sights from empirical evidence. U. S., Technical report. Army Research Inst for the Behavioral and Social Sciences, Orlando, FL (2011)
34. Kelley, C.R.: What is adaptive training? Hum. Factors **11**, 547–556 (1969)
35. Cha, H.J., Kim, Y.S., Park, S.H., Yoon, T.B., Jung, Y.M., Lee, J.-H.: Learning styles diagnosis based on user interface behaviors for the customization of learning interfaces in an intelligent tutoring system. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 513–524. Springer, Heidelberg (2006). https://doi.org/10.1007/11774303_51

36. Khalil, H., Ebner, M.: MOOCs completion rates and possible methods to improve retention - a literature review. In: EdMedia: World Conference on Educational Media and Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications, AACE, Chesapeake, VA (2014)
37. Bellman, R., Kalaba, R.: On adaptive control processes. IRE Trans. Autom. Control **2**(4), 1–9 (1959)
38. Hull, C.L.: Mind, mechanism, and adaptive behavior. Psychol. Rev. **44**, 1–32 (1937)
39. van Oijen, J., Roessingh, J.J., Poppinga, G., Garcia, V.: Learning analytics of playing space fortress with reinforcement learning. In: Sottilare, R., Schwarz, J. (eds.) HCII 2019. LNCS, vol. 11597, pp. 363–378. Springer, Cham (2019)
40. US Army, Field Manual No. 3-21.8 - The Infantry Rifle Platoon and Squad. Department of the Army, Washington, D.C., 28 March 2007