

A Neural-Symbolic System for Automated Assessment in Training Simulators

A Position Paper

Leo de Penning, Bart Kappé, Karel van den Bosch

TNO Defense, Security and Safety

Soesterberg, The Netherlands

leo.depenning@tno.nl, bart.kappe@tno.nl, karel.vandenbosch@tno.nl

Abstract

Performance assessment in training simulators is a complex task. It requires monitoring and interpreting the student's behaviour in the simulator using knowledge of the training task, the environment and a lot of experience. Assessment in simulators is therefore generally done by human observers. To capture this process in an automated system is challenging and requires innovative solutions. This paper proposes a new module for automated assessment in simulators that is based on Neural-Symbolic Learning and Reasoning and the Recurrent Temporal Restricted Boltzmann Machine (RTRBM). The module is capable of using existing and learning new rules for performance assessment, by observing experts and students performing the training tasks. These rules are used to validate and support the assessment process and to automatically assess student performance in a training simulator. The module will be developed in a three year research project on assessment in driving simulators for testing and examination.

1 Introduction

Performance assessment in training simulators has always been a complex task that is generally performed by human observers. Performance assessment by automated systems is often limited to simple training tasks, because assessing complex tasks requires the modelling of all interrelations between the information present in the simulation, the training tasks, and the constructs being assessed (e.g. competences). Also, when it comes to more subjective assessments (e.g., how 'safe' is the student driving), conventional modelling techniques fall short, as the applied assessment rules are often implicit and difficult to elicitate from the simulation or domain experts.

We propose a new module for automated assessment as part of the Virtual Instruction platform SimSCORM [Penning *et al.*, 2008]. This assessment module will be able to learn new rules from the task description, (real-time) simulation data, related assessment data of domain experts or students and already existing rules (also called background knowledge). These rules can be presented in a human-

readable ('symbolic') form, facilitating the validation of the assessment rules and supporting the assessment process.

2 Global Architecture

The automated assessment module requires real-time interaction with the simulator(s), the student and human assessors, and a description of the training task, a student profile and the simulated environment. SimSCORM provides a generic platform for definition and presentation of simulation based training content and interaction between the content, its users and the simulation based on international standards (e.g. SCORM, HLA, XML, etc.). Via this platform the automated assessment module can easily access the objects and attributes in the simulation and get information on the student profile and progress.

Figure 1 depicts the automated assessment module (named CogAgent) in the SimSCORM context. SimSCORM provides a player that presents a SCORM based training task to the students and possibly one or more assessors (e.g. teachers, examiners or students) via a (web-based) Learning Management System. This player uses SimAgent to interact with the simulator(s) and CogAgent to do automated performance assessment and learn new assessment rules from observation. Therefore, the player configures CogAgent with information on the training task, measured variables, student profile, assessed constructs and existing symbolic rules. During execution of the training task, assessors can provide feedback on the assessed constructs which will be presented to CogAgent as short-term evaluations (depicted as assessment data). SimAgent will act as a generic interface between the simulator(s) and CogAgent, and pre-processes received data from the simulator(s) based on measured variable descriptions. Based on the information from the player and SimAgent, CogAgent determines an overall (or long-term) evaluation for the assessed constructs which will be presented to the students (and assessors) as assessment result. Parallel to this it uses the measured data and assessment data to adapt the internal knowledge on assessment rules, resulting in new rules that can be validated afterwards. All information, including the symbolic rules, will be encoded in XML as part of the working memory of the agents and will be distributed via SOAP (either locally or via a web-service).

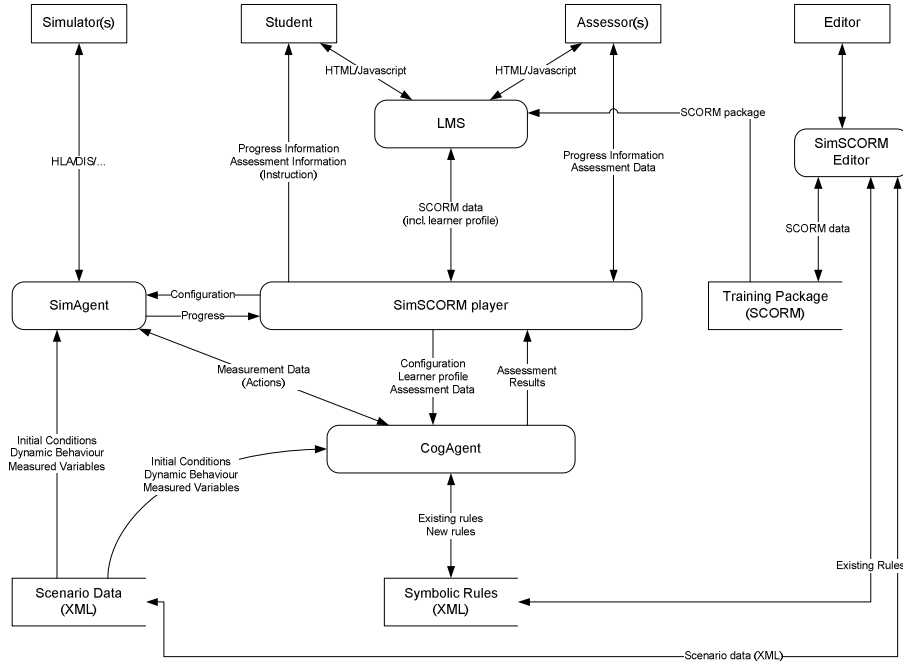


Figure 1. Global architecture of the automated performance assessment module

3 Neural-Symbolic Learning and Reasoning

The CogAgent must be able to learn new rules from observation and existing rules, infer conclusions from these rules and present them in a human readable form. Research on Neural-Symbolic Learning and Reasoning focuses on the integration of learning techniques and architectures from Neural Networks with the symbolic presentation and reasoning techniques in (Fuzzy) Logic Programs (see [Bader and Hitzler, 2005]).

The Neural-Symbolic model proposed for CogAgent is based on the Recurrent Temporal Restricted Boltzmann Machine (RTRBM) [Sutskever *et al.*, 2009] and is depicted in Figure 2. This partially connected symmetric neural network implements an auto-associative memory of its input layers (called visible layers). CogAgent contains three visible layers that represent its beliefs, desires and intentions (introduced by [Bratman, 1999]). Beliefs are variables related to the training task (initial conditions, dynamic behaviour and measured variables) and the student profile. Intentions are variables related to actions or instructions. And desires are variables related to performance assessments (e.g. evaluations or rewards). Beliefs and intentions are directly related to the current state of the context whereas desires will be related to future states as well using Temporal Difference learning [Sutton, 1988]. This technique learns the model to predict a maximum obtainable value for its desires (e.g. overall evaluation scores) based on the current and previous states. Otherwise, the model would only learn to map short-term evaluations, which is not desired in this case.

The hidden layer of the RTRBM is connected to the visible layers with symmetric connections. Each hidden unit represents a rule or relation between one or more visible units. It also contains recurrent hidden-to-hidden connections that enable the RTRBM to learn the temporal dynamics in the visible layers using an algorithm based on contrastive divergence and backpropagation through time. Using this layer we can infer the posterior probability of beliefs, intentions and desires in relation to the state of current and previous beliefs, intentions and desires.

3.1 Symbolic Rules and Fuzzy Atoms

As described in section 2, the rules CogAgent needs to encode, learn and reason about are relations (or causalities) between XML encoded constructs, which will be called atoms hereafter. An XML based atom describes a belief, intention or desire as a function of measured data from the simulator and/or assessment data from the assessors (or students). In case of training simulators this data is often expressed in both continuous and binary values. Therefore we need to use functions in the visible units that can express both. In [Chen and Murray, 2003] sigmoid functions are introduced that contain a ‘noise-control’ parameter to allow a smooth translation from noise-free deterministic behaviour to binary-stochastic behaviour. These continuous stochastic functions can express both binary and continuous variables. The ‘noise-control’ parameter controls the steepness of the sigmoid function and can be trained, such that the behaviour of a function dynamically changes according to the distribution of its input values. We will extend our model with such functions to create a Recurrent Temporal Continuous Restricted Boltzmann Machine (RCTRBM).

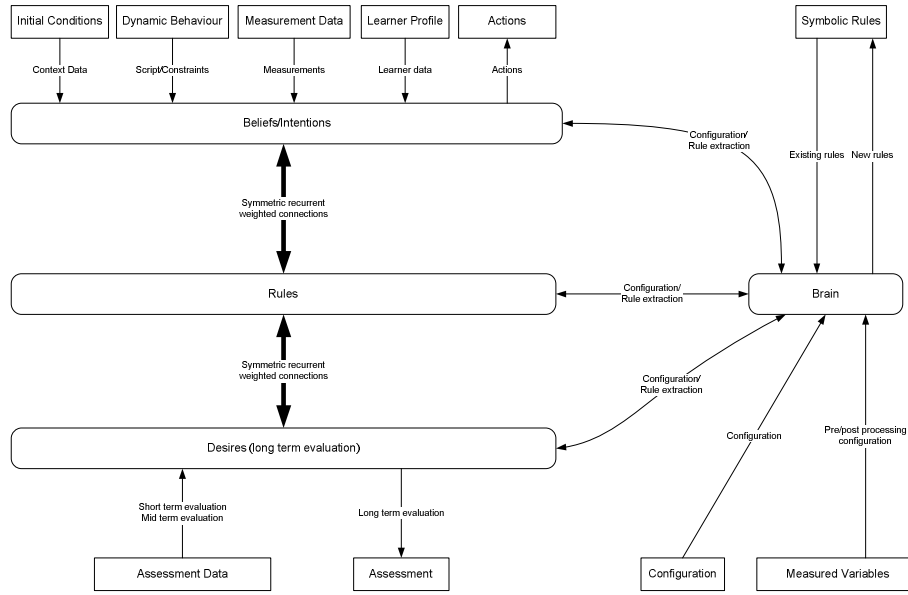


Figure 2. Neural-Symbolic Cognitive Architecture for CogAgent

To express relations between atoms in symbolic rules we propose to use the temporal propositional logic described in Lamb *et al.*, [2007]. This logic contains several modal operators that extend classical modal logic with a notion of past and future. All these operators can be translated to a form that relates only to the immediate previous timestep (denoted by the temporal operator \bullet). This allows us to encode any rule from this language in the RTCRBM as a combination of visible units (or atoms) and recurrent hidden units that represent applied rules in the previous timestep. For example the proposition $\alpha\mathcal{S}\beta$ denotes that a proposition α has been true *since* the occurrence of proposition β . This can be translated to: $\beta \rightarrow \alpha\mathcal{S}\beta$ and $\alpha \wedge \bullet(\alpha\mathcal{S}\beta) \rightarrow \alpha\mathcal{S}\beta$, where α and β are modelled by visible units and $\bullet(\alpha\mathcal{S}\beta)$ is modelled by a recurrent hidden unit.

We extend this logic with the use of equality and inequality formulas to represent the atoms for continuous variables (e.g. $A=x$, $A<x$, etc). Note that the atoms for binary variables can also be represented as $A=true$ or $A=false$, which allows us to handle the outcome of these atoms in the same way as with the continuous atoms. But for readability we will use the classical notion A and $\neg A$.

Due to the stochastic nature of the sigmoid functions used in our model, the atoms can be regarded as fuzzy sets with a Gaussian membership function. This allows us to represent fuzzy concepts, like good and bad or fast and slow or approximations of learned values, which is especially useful when reasoning with implicit and subjective rules. In fact our model can be regarded as a neural-fuzzy system similar to the fuzzy systems described in [Kosko, 1992] and [Sun, 1994].

Now let's take the training task depicted in Figure 3. Using our extended temporal propositional logic, we can describe rules about the conditions, scenario and performance assessment related to this task.

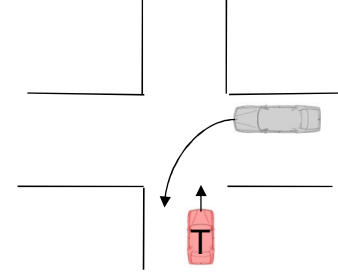


Figure 3. Example training task for driving simulation. The Trainee drives on an urban road, approaching an intersection. The Trainee has to apply the yield-to-the-right-rule.

Example rules for a driver training task:

Conditions:

- (Area = urban)
- (Weather \geq good)
- (Time \geq 6) \wedge (Time \leq 18)

Scenario:

- (Speed > 0) \wedge ApproachingIntersection \rightarrow CrossIntersection
- ApproachingIntersection \wedge \diamond (ApproachingTraffic = right)
- ((Speed > 0) \wedge (HeadingIntersection)) \mathcal{S} (DistanceIntersection < x) \rightarrow ApproachingIntersection

Assessment:

- ApproachingIntersection \wedge (DistanceIntersection = 0) \wedge (ApproachingTraffic = right) \wedge \square (Speed = 0) \rightarrow (Evaluation = good)
- ApproachingIntersection \wedge (DistanceIntersection = 0) \wedge (ApproachingTraffic = right) \wedge \diamond (Speed > 0) \rightarrow (Evaluation = bad)

The rule with temporal operator \mathcal{S} , denotes that *ApproachingIntersection* is true when the driver has been driving towards an intersection since a certain distance x to an inter-

section was passed. This rule and the actual value for x can be learned from observation by clamping the actual speed, heading and distance to the visible units and the value *true* to the unit for *ApproachingIntersection* when the trainee is approaching the intersection. This can be done by an assessor or the student, but could also be automatically inferred by the model, as explained in the next section.

3.2 Rule encoding and extraction

To encode and extract symbolic rules in symmetric connectionist networks, like the RBM, Pinkas [1995] describes a generic method that directly maps these rules to the energy function of such networks. Therefore he describes an extension to propositional logic, called penalty logic that applies a penalty to each rule. This penalty can be regarded as the “certainty” or “reliability” of a rule and is directly related to the weights of the connections between the units that form this rule. To apply the encoding and extraction algorithms of Pinkas successfully to our model we need extend our temporal propositional logic with the use of penalties. [Sun, 1994] describes a method to map atoms with classical modal operators to real values. We propose to extend this method to create a mapping of atoms and rules with the modal operators used in our model to penalties. Furthermore we need to investigate what changes are required to the algorithms to handle the use of equality formulas and continuous variables. For example, we need to prove that it is possible to infer the correct value for unknown continuous variables in a rule via pattern reconstruction based on known values and (previously) applied rules. And to encode and extract rules with inequality formulas we need to be able to transform these to and from rules that contain only equality formulas.

The penalties that are encoded or learned by our model can be used to rank the rules according to their applicability in a certain context or scenario, giving the students and assessors a nice overview of the applied rules. Also they allow us to solve ambiguities in the application of rules, by using such a ranking to select the most applicable (or reliable) rule in each case.

4 Further Research and Experiments

The model described here is still conceptual and requires further research. To summarize the previous sections, we need to investigate the following topics:

- Is the proposed language for symbolic rules adequate enough to represent the subjective and fuzzy rules applied in performance assessment?
- How to determine the penalties of atoms and rules based on their modalities? And how to map penalties to temporal modalities of rules and atoms?
- How to transform rules with inequality formulas to and from rules with only equality formulas?
- If and how to adapt the rule encoding and extraction methods of Pinkas [1995] to make them applicable to the RTCRBM?
- How to integrate temporal difference learning in the RTCRBM for long term evaluation of desires?

These and many other topics will be investigated in a three year research project on assessment in driving simulators, carried out by TNO in cooperation with the Dutch licensing authority (CBR), Research Center for Examination and Certification (RCEC), Rozendom Technologies and ANWB driving schools. The resulting automated assessment module will be validated in several experiments on a large student population using multiple commercial driving simulators. If successful, the module will be used to support the Dutch driver training and examination program.

References

- [Bader and Hitzler, 2005] Sebastian Bader and Pascal Hitzler. Dimensions of neural-symbolic integration - a structured survey. In *We Will Show Them: Essays in Honour of Dov Gabbay, Volume 1. International Federation for Computational Logic*, pages 167-194, College Publications, 2005.
- [Bratman, 1999] Michael E. Bratman. *Intention, Plans, and Practical Reason*. Cambridge University Press, June 1999.
- [Chen and Murray, 2003] Hsin Chen and Alan F. Murray. Continuous restricted Boltzmann machine with an implementable training algorithm. In *Vision, Image and Signal Processing, IEE Proceedings*, pages 153-158, 2003.
- [Kosko, 1992] Bart Kosko. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall, 1992.
- [Lamb *et al.*, 2007] Luís C. Lamb, Rafael V. Borges, Artur S. d'Avila Garcez. A Connectionist Cognitive Model for Temporal Synchronisation and Learning. In *Proceedings of the Conference on Association for the Advancement of Artificial Intelligence (AAAI)*, pages 827-832, 2007.
- [Penning *et al.*, 2008] Leo de Penning, Eddy Boot and Bart Kappé. Integrating Training Simulations and e-Learning Systems: The SimSCORM platform. In *Proceedings of the Conference on Interservice/Industry Training, Simulation, and Education Conference (IITSEC)*, Orlando, USA, December 2008.
- [Pinkas, 1995] Gadi Pinkas. Reasoning, nonmonotonicity and learning in connectionist networks that capture propositional knowledge. In *Artificial Intelligence v.77 n.2*, pages 203-247, September 1995.
- [Sutskever *et al.*, 2009] Ilya Sutskever, Geoffrey E. Hinton and Graham W. Taylor. The Recurrent Temporal Restricted Boltzmann Machine. In *Advances in Neural Information Processing Systems 21*, MIT Press, Cambridge, MA, 2009.
- [Sun, 1994] Ron Sun. A neural network model of causality. In *IEEE Transactions on Neural Networks, Vol. 5, No. 4*, pages 604-611. July, 1994.
- [Sutton, 1988] Richard S. Sutton. Learning to predict by the methods of temporal differences. In *Machine Learning 3*: pages 9-44, erratum page 377, 1988.