From Sensorimotor Graphs to Rules: An Agent Learns from a Stream of Experience*

Marius Raab, Mark Wernsdorfer, Emanuel Kitzelmannn, and Ute Schmid

[1] Cognitive Systems Group, University of Bamberg, Germany
{marius.raab,mark.wernsdorfer,ute.schmid}@uni-bamberg.de
http://www.uni-bamberg.de/kogsys
[2] International Computer Science Institute, Berkeley, USA
emanuel@icsi.berkeley.edu

Abstract. In this paper we argue that a philosophically and psychologically grounded autonomous agent is able to learn recursive rules from basic sensorimotor input. A sensorimotor graph of the agent's environment is generated that stores and optimises beneficial motor activations in evaluated sensor space by employing temporal Hebbian learning. This results in a categorized stream of experience that feeds in a Minerva memory model which is enriched by a *time line* approach and integrated in the cognitive architecture Psi—including motivation and emotion. These memory traces feed seamlessly into the inductive rule acquisition device Igor2 and the resulting recursive rules are made accessible in the same memory store. A combination of cognitive theories from the 1980ies and state-of-the-art computer science thus is a plausible approach to the still prevailing *symbol grounding problem.*

Keywords: symbol grounding, temporal Hebbian learning, cognitive architecture, inductive rule learning

1 Introduction

"How can the semantic interpretation of a formal symbol system be made *intrinsic* to the system, rather than just parasitic on the meanings in our heads?" Since Harnad [7, p.335] has posed this question, the *symbol grounding problem* is an ongoing issue in AI. Progress has been made, but the problem is not solved [22].

We argue that *old-fashioned* cognitive theories from the 1980ies together with state-of-the-art learning systems allow for bridging the gap between symbolic and sub-symbolic approaches: a dichototmy that is present even in state-of-the-art architectures like Clarion [23]. Here, we meet with Langley et al. [13, p.155f] who urge for "more research on architectures that directly support both episodic memory and reflective processes that operate on these structures it contains."

In our view mental competencies evolve from a structural coupling to the world (outside and inside) [28]—in contrast to predefined competencies which

* This is an author-created version of the published article. The final publication is available at www.springerlink.com

cannot be said to depend on anything else than the architect's beliefs [7]. Not until mental representations are justified by referring to relevant and meaningful entities, abstract concepts can be inferred.

The generation of abstract concepts is an ultimate touchstone for a system. Here, abstract concepts are recursive, i.e. infinite, regularities. This goes beyond Anderson's [1] deductive and abductive rule-generation; as, for example, deduction always raises the question where the premises come from. It also goes beyond systems like Clip [2] and Clarion [23,24]—these systems lack the ability to cope with recursive regularities—a concept like *odd/ even* is out of their scope.

In the following we present our approach to learn rules from streams of experience which is composed of three succeeding layers: In a first step (Sect. 2) a continuous sensorimotor space is segmented by constructing prototypes based on an evaluation function which balances exploitation and exploration. In a second step (Sect. 3) graphs containing these prototypes are enumerated along a time line, associated with the reward experience and transformed into simple symbolic rules. Finally (Sect. 4), regularities in these simple rules are detected and the rules are folded into a recursive rule set which generalizes over the previous experiences.

The overall system controls a virtual autonomous agent (AA) which moves in a *Discworld*. An even number of circles means that the agent receives reward at the innermost spot, an odd number evokes punishment. The AA has to avoid harmful targets and approach desirable ones. All necessary knowledge transformations in our agent are done syntactically, so no additional *meaning* is introduced on the way.

2 Learning Context-Sensitive Partitioning of Sensorimotor Space

The AA features a set of sensors and actors. In our simple *Discworld* scenario the sensors detect changes in brightness and the motors power a differential drive. Initially, the environment is explored by random walk and later guided by previous experience trying to repeat beneficious actions and avoid harmful ones. Experience is represented in sensorimotor vectors which are integrated in a graph.

The sensorimotor vectors define Voronoi cells and thereby generate a segmentation of the otherwise continuous sensorimotor space [27]. This characteristic is exploited for categorization, that is construction of abstract prototypes [25] representing collections of experiences. Each prototype holds the activation perceived and the evaluation received during its creation.

Temporally successive prototypes are connected to form a graph representing possible next states. This information can be used to predict possible outcomes of actions or perceptions. New experiences are evaluated with respect to their similarity to already existing prototypes.

By employing temporal Hebbian learning [19], the weighted edges between prototypes are reinforced if memorised sequences are confirmed by the envi-

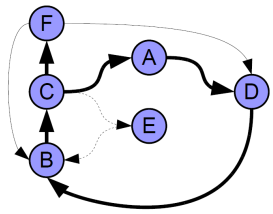

Fig. 1. Saturated motor graph

ronment. Alternative connections are inhibited by weakening all contradicting outgoing connections from the previous prototype and all contradicting incoming connections to the momentary prototype. Dropping below a critical threshold, connections are eventually removed from the graph. A mature sensorimotor graph (see Fig. 1), allows for anticipating (con-) sequences of arbitrary sensorimotor activations. By removing connections, the graph fragments into isolated components containing only a subset of sequential prototypes.

The activation vectors stored in prototypes are *exploited* by random optimization. The evaluation gradient dropping below zero is interpreted as indication for having reached a local maximum. Therefore, by performing a 1-nearest-neighbour query within the active graph, the algorithm tries to switch to the next best prototype representing the present sensorimotor activation. In case the present prototype is returned anew, the algorithm tries to *explore* new actions by creating a new prototype.

After a test run one graph component has been extracted. We visualised the graph in Fig. 1, where thicker edges representing stronger connections. A circular path within the motor graph indicates successful coping with the environment. Here this path of continuously reinforced edges is $C \rightarrow A \rightarrow D \rightarrow B \rightarrow C$. Once the graph components are mature and stable, switching between them can be regarded as changing situational context. These components are grounded representation that serve as entities for further processing. Their creation takes place as follows:

1. The agent learns *symbols* bearing *relevancy* and *meaning*.
2. Thereby the agent segments the sensorimotor space according to [15].
3. A *context* component emerges like: *line – line – line – reward* or *line – line – punishment*

3 Handling a Stream of Experience

Feeding these components into a psychologically valid model—by integrating the Psi-theory of Dietrich Dörner [6], and by utilizing the Minerva memory model by Hintzman [9]—we are able to induce explicit rules and to feed these rules back into the agent's memory.

We chose the cognitive architecture Psi by Dörner ([6], see also [3]) as a framework. Dörner postulates a *protocol memory* as the only memory structure—i.e., a chain of perceptual input, motor actions and changes in motivational states. Perception, for example, is a constant matching of input against stored events, where missing or ambigious inputs are supplemented by stored fragments (schemata). But how exactly to build such a structure remains vague. A similiar idea of a single-store system was voiced by Klahr & Wallace ([11, p.366], refined in [26]). They assume the existence of "an ordered list representing the relative values as a result of the child's experience to date".

The Minerva 2 framework by Hintzman [9] is in some aspects similiar: "[. . .] a vast collection of episodic memory traces, each of which is a record of an event or experience [. . .] Minerva 2 represents an attempt to account for data from both episodic and generic memory tasks within a single system." (p.96) This contrasts, for example, with Soar where episodic memory is a collection of *snapshots* of working memory, and where procedural and semantic memory are further compartments [12].

In Minerva, *any* event will result in a new memory trace, no matter if a very similiar event has been experienced before or not. There is no chronological, overall time line any more; instead a huge set of seperate traces is built. Memory traces are configurations of primitive properties, like sensory features; simply modelled as vectors filled with *1*, *0* and *-1* values. Categories and abstract knowledge are built at the time of retrieval and not during encoding and storage. As Dougherty [5] has shown, Minerva lends itself to extensions covering more complex aspects of reasoning. Motivational states sensu Dörner could easily be represented by additional vectors.

We propose a framework where information will be stored in a single system, in chronological order, with regularity detection as core learning mechanism. Nodes in this model are simple feature vectors. This combination merges a parsimonious yet potentially powerful cognitive-emotional-motivational approach with a simple yet powerful learning concept and a frugal, empirically founded memory model. We will show how an inductive learning approach fits in such a framework—and how it takes our system to the next level while at the same time staying *grounded*.

4 Learning Productive Rules

The sketched model will be the necessary prerequisite for integrating the inductive rule acquisition device Igor2 [10]. This machine learning device has already been used successfully for classical cognitive tasks like the *Tower of Hanoi* [21]. In short, it is a means to construct recursive functional programs from a few non-recursive examples. However, so far this learning device was not embedded into a psychological model.

Igor2 combines analytical recurrence detection with a guided search in program space. Programs are represented as *constructor systems*. As with every learning system, Igor2 relies on some biases: Programs must be a valid subset of HASKELL, search space ist explored by preferring a minimal number of case distinctions, and the given input patterns must not unify pairwise. Furthermore, for a given induction problem the *first n examples* must be given.

Being part of a larger cognitive architecture, we deem this inductive device a valid cognitive approach. IGOR2 abstracts from given examples by generating a *least general generalisation*, partitions the problem space, treats sub-problems as new problems and uses previously learned rules. These are plausible cognitive mechanisms (see, e.g., [18])—and, in any case, much closer to human reasoning than, for example, generate-and-test learning devices. We used the inductive rule acquisition device IGOR2 [10] to learn knowledge-level production rules from basic sensorimotor data.

How to build rather complex input rules in HASKELL-compatible syntax without semantic assumptions; in other words, without instructions that are "parasitic to the system" in the sense of [7]? In a proof-of-concept-implementation (in JAVA), we did so using a cascade of mere syntactical transformations:

4. A bijective mapping on ASCII characters retains the distinction of context components, introduces no new information, and results in: $LLL \rightarrow TRUE$ or $LL \rightarrow FALSE$. We chose this kind of transformation, as an ASCII character can be seen as a 8-bit feature vector in the sense of MINERVA. By arbitrary combination of characters, the feature vector's length is unrestricted.
5. The resulting set of rules is fed into IGOR2
6. The output is a recursive rule:
   $learn[] = False; \quad learn['L'] = True; \quad learn('L' : ('L' : a0)) = learn\ a0$
7. Such a rule can easily and automatically be transformed into a grammar:
   $S \rightarrow\ X \qquad X \rightarrow\ 'L' \qquad X \rightarrow\ 'L''L'X$
8. Consecutively, standard algorithms (like in JFLAP [20]) might be used to syntactically transform this grammar into a nondeterministic finite automaton.
9. This automaton represents the recursive part of the regularity; and the nodes still hold the sensory information associated from the grounding process. When this fragment is connected to existing memory (here: the strict linked list is extended to hold several connections, resulting in a graph), it will be searchable by MINERVA probes, too.

5 Conclusion and Future Work

Right now no motivation and emotion is implemented. From a theoretical viewpoint, however, the agent is prepared. And planning, as described by Dörner, can take place. Also, there's no mechanism to remove rules from memory. This would be necessary before these rules would enter a planning module.

The agent presented bears potential that was not intended in the first place. The MINERVA representation, for example, lends itself to analogical reasoning. As 1 codes a necessary feature, −1 a prohibitive one, and 0 a 'don't care', cross-context learning would be a logical AND over two situations (i.e., vectors).

The presented system might behave strangely, irrational or superstitious. Newberg et al. [18] describe how cognition could be seen as the core of belief, myth and religion. Maybe, the AA proposed here has the potential to create such a form of *meaning* for itself, based on its own grounded symbols. Of course, like Winograd & Flores [28] note, this agent would not be human. Yet, it could be a bit more *like* a human.

References

1. Anderson, J. R.: How can the human mind occur in the physical universe? Oxford University Press, Oxford (2007)
2. Avila Garcez, A. S., Zaverucha, G.: The Connectionist Inductive Learning and Logic Programming System. Applied Intelligence 11, 59–77 (1999)
3. Bach, J.: Principles of Synthetic Intelligence: PSI: An Architecture of Motivated Cognition. Oxford University Press, Ofxord (2009)
4. Butz, M.: Self-organizing sensorimotor graphs plus internal motivations yield animal-like behavior. Adaptive Behavior 18, pp. 315 (2010)
5. Dougherty, M. R. P., Gettys, C. F., Ogden, E. E.: MINERVA-DM: A Memory Processes Model for Judgments of Likelihood. Psych. Review, 106(1), 180–209 (1999)
6. Doerner, D.: Bauplan für eine Seele. Rowohlt, Reinbek (1999)
7. Harnad, S.: The Symbol Grounding Problem. Physica D, 42, 335–346 (1990)
8. Heidegger, M.: Sein und Zeit. Niemeyer, Tübingen, 16th edition (1986)
9. Hintzman, D. L.: MINERVA 2: A simulation model of human memory. Behavior Research Methods, Instruments & Computers, 16(2), 96–101 (1984)
10. Kitzelmann, E.: A Combined Analytical and Search-Based Approach to the Inductive Synthesis of Functional Programs. PhD thesis, University of Bamberg (2010)
11. Klahr, D., Wallace, J.: An Information Processing Analysis of Some Piagetian Experimental Tasks. Cognitive Psychology, 1, 358–387 (1970)
12. Laird, J.E.: Extending the Soar cognitive architecture. In: Proceedings of the artificial general intelligence conference. Memphis, TN (2008)
13. Langley, P., Laird, J. E., Rogers, S.: Cognitive architectures: Research issues and challenges. Cognitive Sytems Research 10(2),141–160 (2009)
14. Matyas, J.: Random Optimization, In: Automaton and Remote Control (26 (2)), 246–253 (1965)
15. Mayo, M.: Symbol grounding and its implications for artificial intelligence. In: Proceedings of the 26th Australasian computer science conference-Volume 16 , Australian Computer Society, Inc., 55–60 (2003)
16. Mitchell, T. M.: Machine learning. McGraw-Hill, Boston, MA (1997)
17. Nadel, L.: Distributed Representations. In: C. R. Encyclopedia of cognitive science, nature publishing group (2003)
18. Newberg, A., d'Aquili, E., Rause, V.: Why God Won't Go Away: Brain Science and the Biology of Belief. Ballantine Books, NY (2002)
19. Porr, B. and Wörgötter, F.: Temporal Hebbian learning in Rate-Coded Neural Networks: A theoretical approach towards classical conditioning. In: Artificial Neural Networks – ICANN 2001', Springer, 1115–1120 (2001)
20. Rodger, S. H.: JFLAP. An interactive formal languages and automata package. Jones and Bartlett's, Sudbury, MA (2006)
21. Schmid, U., Hofmann, M., Kitzelmann, E.: Analytical inductive programming as a cognitive rule acquisition devise. In: Proceedings of the Second Conference on Artificial General Intelligence, pp. 162–167, Amsterdam. Atlantis Press (2009)

22. Steels, L.: The Symbol Grounding Problem Has Been Solved. So What's Next? In: de Vega, M. (ed.), Symbols and Embodiment: Debates on Meaning and Cognition, ch. 12. Oxford University Press, Oxford (2008)
23. Sun, R.: The CLARION cognitive architecture: Extending cognitive modeling to social simulation. In: Sun, R. (ed.), Cognition and Multi-Agent Interaction. Cambridge University Press, New York (2006)
24. Sun, R.: The motivational and metacognitive control in CLARION. In: Gray, W. D. (ed.), Integ. Models of Cog. Systems. Oxford University Press, New York (2007)
25. Vanpaemel, W. and Storms, G.: In search of abstraction: The varying abstraction model of categorization. In: Psychonomic bulletin & review (15:4), pp. 732 (2008)
26. Wallace, J., Klahr, D., Bluff, K.: A Self-Modifying Production System Model of Cognitive Development. In: Klahr, D., Langley, P., Neches, R.T. (eds.): Prod. system models of learning and development, pp. 359–435. MIT Press, Cambridhge, MA (1987)
27. Watson, D.: Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. In: The computer journal (24:2), pp. 167 (1981)
28. Winograd, T., Flores, F.: Understanding Computers and Cognition. A New Foundation for Design. Ablex Corporation, Norwood, NJ (1986)