Knowledge Self-Adaptive Multi-Agent Learning

Simon Reichhuber[1]

Abstract: In this paper concepts of a starting Doctoral Dissertation are presented, discussing the question how agents constructed according to Organic Computing methodologies can autonomously identify Knowledge Sources and adapt them to their learning procedure. Achieving this, the fields of Multi-Agent Learning, Organic Computing, Transfer Learning, and Online Learning are combined to an unified architecture. The focus of the work is on the real-time evaluation of knowledge sources. In order to show the practical use case of such systems, the author presents two scenarios. The first, collaborative crawling, is an information retrieval task, hence it deals with knowledge distributed over multiple websites. Whereas the latter is designed to run in a virtual space, the second, denoted as machine park collaboration, can be implemented in industrial 4.0 fields of the real world.

Keywords: Organic Computing; Online Learning; Multi-Agent Learning; Transfer Learning

1 Introduction

Nowadays the prevalence of sensor systems, for instance in the area of wearable devices [MB09] and smart homes [**CYHG03**], have confirmed the vision of Ubiquitous Computing [We93] presented by Mark Weiser in 1993. According to Weiser, "computers should be autonomous agents that take on our goals". These agents has been developed further in the domain of Organic Computing (OC) [MT17] investigating self-* properties, like self-optimisation, self-organisation, or self-adaption. Given contextual data, an agent decides how to react on this information. Naturally, the question arises of how an agent adapts this data to its learning, since there might be unrelevant data which can be excluded from learning. In the perspective of an agent that is equipped with an individual goal it might be reasonable to adapt its empirical inputs. Not only static data from databases or sensors but also other agents might be valuable for the training. Inspired by [Ca17], where the authors describe paradigms to systems self-reflecting their learning behaviour, the author denotes any collection of data retrievable by a local interface Knowledge Source (KS). How agents deal with such KS in a dynamically changing enverinment will be investigated in the scope of the doctorial thesis and in the further of this paper, which is organised as follows: Starting with a brief motivation by identifying a knowledge gap in the related literature in Sect. 2 regarding autonomously knowledge-source-selecting agents. In Sect. 3 the architecture of such systems is presented. Under this circumstances we formulate open research questions in Sect. 4. Then, in Sect. 5 the basic architecture of an autonomous and knowledge source self-adapting learning agent is applied on two applications. Finally, we conclude the paper and give an outlook of the further work in Sect. 6

[1] Chair of Intelligent Systems, University of Passau, Passau Germany simon.reichhuber@uni-passau.de

doi:10.18420/inf2019_ws54

2 Related literature

Throughout the literature of Multi-Agent Reinforcement Learning, there are proposed agents that are either adapting their learning [BV02], cooperating together [CB98], dealing with multi-tasks [Wi07] transferring knowledge autonomously [TKS08], or more specific in a bidirectional way [Ta13] but no approach concerns all the aspects in combination. Transferring knoweledge from another agent, Taylor formulated three steps [TKS08] that an agent consecutively has to be fulfilled: First which task, also known as source task he might select to transfer from. Second, how are its task and the source task are related with? Third, how can an agent transfer the identified knoweledge from source task to target task effectively? From a more general view, similar questions were also asked in the paper of Calma et al. in 2017 [Ca17]. Concerning the more general knowledge source, instead of source tasks from agents only, the authors deal also with statical databases and user queries. They list the currently available techniques of Machine Learning (ML) and describe paradigms of how to tackle this challenge. Though the author is inspired by this paper and the variaty of the listed ML techniques, he has to narrow the field and restrict to a certain architecture to provide practical results. In the following the architecture will be described in more detail.

3 Architecture of autonomously knowledge-source-adapting agents

This section offers a blueprint for the implementation endowing agents with the capabilities of self-optimising their learning. There are several possibilites of introducing an architecture for agents that reflect their learning. Motivating a system of agents that autonomously reaching a goal, the author choose the context of Multi-Agent Reinforcement Learning (MARL) [BBD+08; SPG03] as it provides a general approach of achieving the goal in a greedy way by agents maximizing their reward earned in every single timestep. However the construction of such a reward function for KS-adaptive agents might be more complex and prophetical, since agents that are learning bad expirience does not mean they take immediately the false actions but maybe in a long-term run. To guarantee the required "deepnessöf the described agent, the three-layered OC learning approach [To11] was found to be suitable. On level 0 in Figure 1 there is a so called System Under Observation and Control (SUOC). In this context a reinforcement learning agent can be seen as a System under Observation and Control (SuOC). The higher level 1 is denoted as online-learning level where an Observer/Controller (O/C) pattern [Ri06] monitors the performance of the reinforcement learner and adapts KS with the means of a rule-based learning system, defined in [Wi95] as extended Learning Classifier System (XCS). One layer above on level 2, another O/C observes the layers below, selects new KSs, evaluates all available KSs, and also provides system optimisation by applying Genetic Algorithms. Finally, the layer 3 manages the user requests about the current goal of the agent and also communicates with other agents and requests for certain KSs.

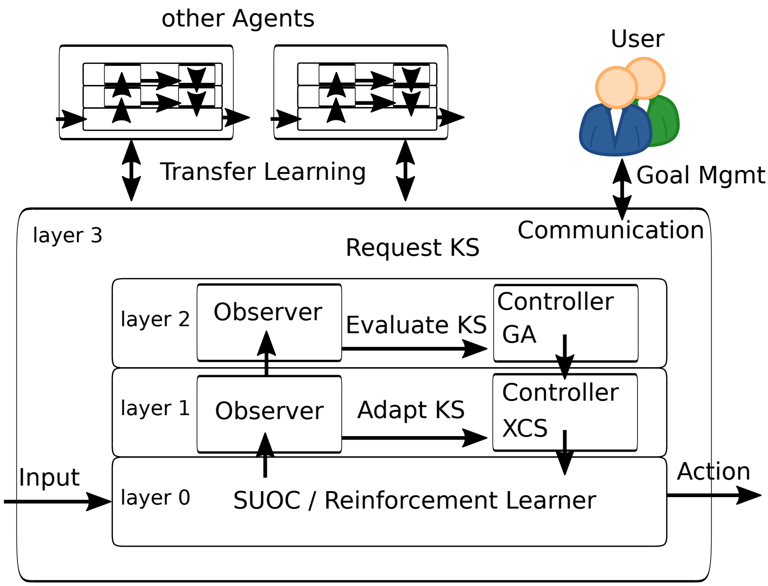

Fig. 1: Adaptive-learning agent architecture.

4 Research Questions

Before formulating research questions, it is helpful to identify involved entities and in which roles they are able to slip into. There are three major entities, the environment, the agents, and the knowledge sources, whereas agents can act as knowledge source but not vice versa. Given a domain or scenario, the environment and knowledge sources are predefined and rather passive. Contrarily, the agents apply actions on the environment and interactions between other agents.

4.1 Learning the representation of knowledge

How to learn attributes of knowledge sources with the properties found in [Ca17]:

- Spatial attributes
  - Locality
- Temporal attributes
  - Spontaneous occurrence
  - Temporary availability
- Quality

  - Quality of Service
  - Level of expertness
- Costs
  - Transaction costs
  - Resources

to predict their behavior in order to adapt them dynamically to a multi-agent learning system?

4.2 Get to know the source

How to evaluate the learned knowledge representation? Which evaluation mechanism is capable of evaluating the source online?

4.3 Learning from other agents

By using techniques from the domain of Transfer Learning [La12; TS09], agents exchange knowledge with the means of a communication protocol. Therefore the agents must proceed the following tasks:

Allocation Initially, the agent has to scan its neighborhood and find all potentially collaborating agents.

Estimated collaboration benefit Then, the agent estimates the use of a collaboration with one or multiple agents. For this purpose, they take their individual learning objective into account and compare it with others. Only if a certain similarity value is observed, the improvement of knowledge compensates the collaboration effort, i.e. formulating knowledge requirement message, waiting for response, and load responded knowledge into learning model.

Adaptation If the collaboration is worth it, the knowledge can be exchanged. The learning will be adapted according to the novel knowledge.

Actual collaboration benefit After testing the learning capabilities by means of common Machine Learning models, like the f1-score, the estimation procedure can be adapted. This step influences the collaboration level of the agent. The higher the estimated costs of collaboration are estimated, the less likely an agent will collaborate with other agents.

4.4 Allocation of knowledge sources to agents

When to adapt knowledge sources to the learning agents? How to perform self-performance measurements? How to match knowledge sources to learning techniques?

5 Applications in distributed domains

In this section possible scenarios are described, for which the knowledge self-adaptive multi-agent learning system can be integrated and evaluated.

5.1 Collaborative crawling

In this scenario the agents are usually called crawlers or spiders (see Figure 2) and a knowledge source is associated with a certain Uniform Resource Locator (URL). In this highly dynamic web of URLs, spiders are able to move along hyperlinks to search keywords within the heterogeneous structure of the world wide web. Given a user query, it is the crawler's challenge to find a suitable path collecting the most significant information about the request. In this thesis an approach of distributed crawling should be investigated where multiple crawlers are distributed and run in parallel. During the search they can exchange knowledge, i.e. the topic domain, related, or non-related keywords, improving their route guidance. Performance evaluation might be the number of sites fulfilling the predicates of the query compared to the amount found with a non communicating distributed crawling system. Balasubramanian et al. [BCQ09] contributed an approach minimizing the cross-links and balancing the work and memory costs of a set of URLs and Hyperlinks that is partitioned and each partition is assigned to a lightweight spider. Whereas the partitioning of Balasubramanian et al., or also the approaches of [CC02; GLM06], work on a global scope, our goal is to train autonomous spiders equipped with Learning capabilities like a Learning Classifier System as XCS [Wi95] where the balancing is implicitly taken into account by the cost function of a Multi-Agent Reinforcment Learning (MARL) setting.

5.1.1 Challenges

Communication A communication protocol should guarantee asynchronous communication.

Parallelisation There are two reasonable design decisions, which have to be evaluated. The spiders might be assigned to different cores of a CPU, which facilitates the inter-spider communication effort, since they share the same local memory. On the other hand, a distribution of spiders among multiple devices enable the usage of swarms of spiders. For instance, a swarm of spiders running on a grid.

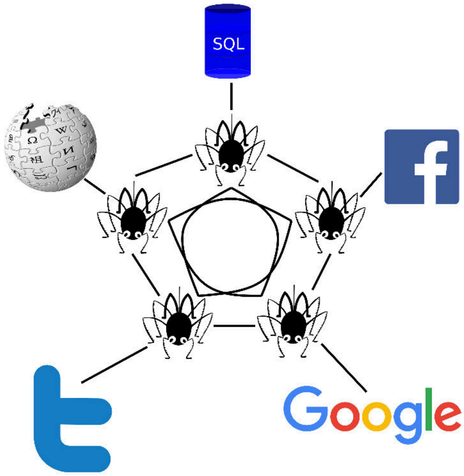

Fig. 2: Sample distribution of autonomous spiders located at different knowledge domains.

Route Guidance In order to improve the route guidance of a single spider, i.e. the subset of hyperlinks of the currently visible hyperlinks available at an URL, it might require information about related keywords, an accurate time range, or geographical limitations.

Spider splitting or merging As the environment in the context of crawling is the web, which is a virtual space, two spiders can easily be joined, since no moving effort is required. The splitting can be based on the number of URLs crawled by two spiders simultaneously. Since the web can be represented as a graph (U, H), where nodes are represented by URLs U and edges by hyperlinks H, two spiders S_1, S_2 crawling on different sub-graphs, (U_1, H_1) and (U_2, H_2) respectively, can be merged to a single spider if they share more than a predefined number of URLs $|U_1 \cup U_2| \geq N_{URL}$.

5.2 Machine park collaboration

A common problem in industry, is the question of how to assign jobs to machines. This problem is also known as the job-shop scheduling problem [Ma60], for which, given machines $M_1, \ldots, M_n$, jobs $J_1, \ldots J_m$ and a cost function $c(M_i, J_j)$ expressing the of running job J_j on machine M_i, it is the task to find a scheduling where each job is assigned to a machine once. As computing the solution is NP-hard [SS95], it is more interesting to adapt this setting to a more practical scenario introducing the following extensions: First, the job list is not fully available, only a window of k jobs can be observed by the machines. Additional jobs will be queued into the list following a FIFO strategy. Second, the machines are parametrized and are able to self-adapt their parametrization, which may influence the costs for certain jobs. Third, the machines can exchange knowledge with each others and a user that can be queried by high costs. The extensions enrich the static problem to a dynamic scenario and the global goal is to maximize the job-finishing rate, rather than finding the optimal schedule of jobs. This scenario is related to the domain of Predictive Maintenance. An general setting is illustrated in Figure 3.

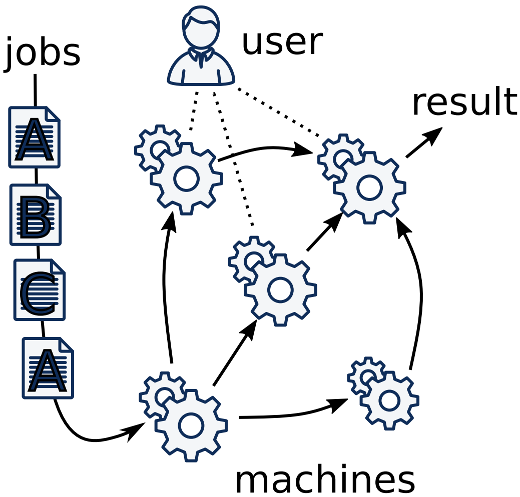

Fig. 3: Machine park visualization with interacting user.

6 Conclusion and future research

The concept of knowledge self-adaptive multi-agent learning is a knowledge-focused approach, where self-learning improvements evolve by an analysis and adaption of empirical knowledge sources. The research questions about knowledge representation, get to know the knowledge source, learning from other agents, and knowledge source self-adapting agents has been investigated in more detail. Further work will concentrate in the resilience and stability over time by implementing and simulating the described scenarios.

References

[BBD+08] Bu, L.; Babu, R.; De Schutter, B., et al.: A comprehensive survey of multiagent reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 38/2, pp. 156–172, 2008.

[BCQ09] Balasubramanian, S.; Chavet, L.; Qi, R.: System, method, and service for collaborative focused crawling of documents on a network, US Patent 7,552,109, June 2009.

[BV02] Bowling, M.; Veloso, M.: Multiagent learning using a variable learning rate. Artificial Intelligence 136/2, pp. 215–250, 2002.

[Ca17] Calma, A.; Kottke, D.; Sick, B.; Tomforde, S.: Learning to learn: Dynamic runtime exploitation of various knowledge sources and machine learning paradigms. In: 2017 IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS* W). IEEE, pp. 109–116, 2017.

[CB98] Claus, C.; Boutilier, C.: The dynamics of reinforcement learning in cooperative multiagent systems. AAAI/IAAI 1998/, pp. 746–752, 1998.

[CC02] Chung, C.; Clarke, C. L.: Topic-oriented collaborative crawling. In: Proceedings of the eleventh international conference on Information and knowledge management. ACM, pp. 34–42, 2002.

[GLM06] Gao, W.; Lee, H. C.; Miao, Y.: Geographically focused collaborative crawling. In: Proceedings of the 15th international conference on World Wide Web. ACM, pp. 287–296, 2006.

[La12] Lazaric, A.: Transfer in reinforcement learning: a framework and a survey. In: Reinforcement Learning. Springer, pp. 143–173, 2012.

[Ma60] Manne, A. S.: On the job-shop scheduling problem. Operations Research 8/2, pp. 219–223, 1960.

[MB09] McCann, J.; Bryson, D.: Smart clothes and wearable technology. Elsevier, 2009.

[MT17] Müller-Schloer, C.; Tomforde, S.: Organic Computing-Technical Systems for Survival in the Real World. Springer, 2017.

[Ri06] Richter, U.; Mnif, M.; Branke, J.; Müller-Schloer, C.; Schmeck, H.: Towards a generic observer/controller architecture for Organic Computing. GI Jahrestagung (1) 93/, pp. 112–119, 2006.

[SPG03] Shoham, Y.; Powers, R.; Grenager, T.: Multi-agent reinforcement learning: a critical survey. Web manuscript/, 2003.

[SS95] Sotskov, Y. N.; Shakhlevich, N. V.: NP-hardness of shop-scheduling problems with three jobs. Discrete Applied Mathematics 59/3, pp. 237–266, 1995.

[Ta13] Taylor, A.; Duparic, I.; Galván-López, E.; Clarke, S.; Cahill, V.: Transfer learning in multi-agent systems through parallel transfer./, 2013.

[TKS08] Taylor, M. E.; Kuhlmann, G.; Stone, P.: Autonomous transfer for reinforcement learning. In: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1. International Foundation for Autonomous Agents and Multiagent Systems, pp. 283–290, 2008.

[To11] Tomforde, S.; Brameshuber, A.; Hähner, J.; Müller-Schloer, C.: Restricted online learning in real-world systems. In: 2011 IEEE Congress of Evolutionary Computation (CEC). IEEE, pp. 1628–1635, 2011.

[TS09] Taylor, M. E.; Stone, P.: Transfer learning for reinforcement learning domains: A survey. Journal of Machine Learning Research 10/Jul, pp. 1633–1685, 2009.

[We93] Weiser, M.: Some computer science issues in ubiquitous computing. Communications of the ACM 36/7, pp. 75–84, 1993.

[Wi07] Wilson, A.; Fern, A.; Ray, S.; Tadepalli, P.: Multi-task Reinforcement Learning: A Hierarchical Bayesian Approach. In: Proceedings of the 24th International Conference on Machine Learning. ICML '07, ACM, Corvalis, Oregon, USA, pp. 1015–1022, 2007, ISBN: 978-1-59593-793-3, URL: http://doi.acm.org/10.1145/1273496.1273624.

[Wi95] Wilson, S. W.: Classifier fitness based on accuracy. Evolutionary computation 3/2, pp. 149–175, 1995.