Learning Classifier Systems for Self-Explaining Socio-Technical-Systems

Michael Heider[1], Richard Nordsieck[2], Jörg Hähner[1]

[1]Organic Computing Group, Universität Augsburg, Germany
[2]XITASO GmbH IT & Software Solutions, Augsburg, Germany
michael.heider@uni-a.de

Abstract

In socio-technical settings, operators are increasingly assisted by decision support systems. By employing these, important properties of socio-technical systems such as self-adaptation and self-optimization are expected to improve further. To be accepted by and engage efficiently with operators, decision support systems need to be able to provide explanations regarding the reasoning behind specific decisions. In this paper, we propose the usage of Learning Classifier Systems, a family of rule-based machine learning methods, to facilitate transparent decision making and highlight some techniques to improve that. We then raise three general research questions that should be answered for any machine learning-based recommendation agent and four additional questions that are more tailored towards rule-based systems. These seven stakeholder-focussed questions provide a template for the approach of self-explaining decision support systems in new domains or settings.

Introduction

Increasing automation of manufacturing creates a continuous interest in properties commonly associated with life-like or organic computing systems, such as self-adaptation or self-optimisation, within the producing industry (Permin et al., 2016). These properties are often achieved using data driven and learning methods (Zhang et al., 2017; Lughofer et al., 2019; Schoettler et al., 2020) as with increasing digitalisation and IoT efforts, data can be collected in large amounts. In modern factories, products are usually inspected by the machines' operators (or specialized quality assurance personnel; we subsume the different roles under 'operator' here for the sake of simplicity) to assess their quality, cf. Figure 1. Recent advances into automated inspection often integrate computer vision-based approaches (Margraf et al., 2017). However, these can be of limited use when quality is not assessable from the surface, e. g. structural or chemical properties that involve laboratory testing. Thus, these systems currently can only partially automate inspection while the conclusions with regards to machine reconfiguration are still reached manually. This requires a large amount of operator knowledge and experience to achieve optimal or even satisfactory results. In settings

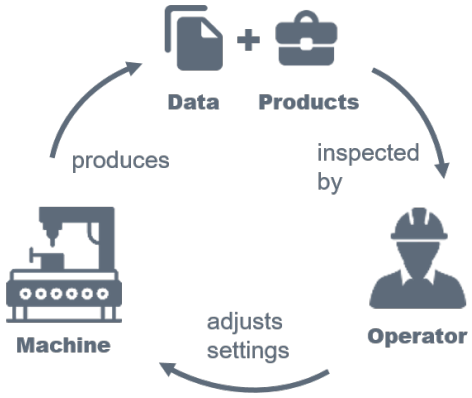

Figure 1: Operator-in-the-loop in today's productions.

with heterogeneous machines and few operators, the strain on operator experience is further increased and production can be seriously threatened by a loss of qualified personnel, e. g. through retirement.

To reduce reliance on specific knowledge of operators and improve the self-adapting and self-optimizing systems, the operator can be assisted by decision support systems. These can easily incorporate large amounts of information simultaneously and are less biased to well known settings, especially compared to operators that only have limited understanding of or experience with the machines. Such decision support systems utilise learning from past experience and ongoing human expert feedback. Combining human operators and supervised learning (SL) agents that collaboratively adjust machines (or lines thereof) manufacturing products expands the socio-technical system with a decision making dimension, cf. Figure 2. Typical shopfloor environments will feature many workers operating on many machines but not necessarily in a one to one array, e. g. multiple workers might be needed to operate a single machine while other machines can be operated by a single worker due to automation. Additionally, to utilise the available data most efficiently, not every machine should need their own model but models should generalise over multiple machines of the same

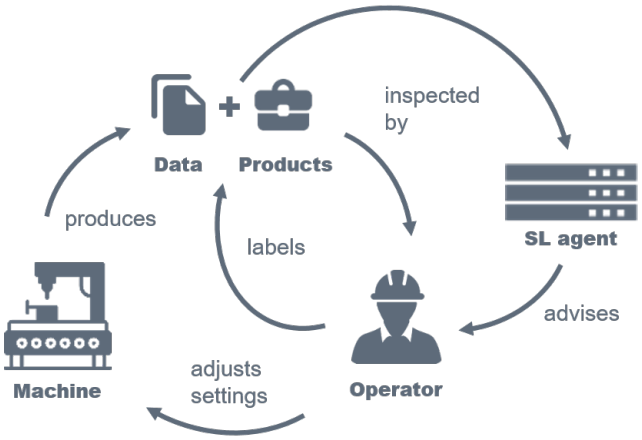

Figure 2: Assisted production using agent trained with supervised learning (SL) during operation.

or similar type. For production lines where multiple models would participate, the parametrisation choices of preceding machines would need to be accounted for by subsequent models, e. g. through the help of models of higher abstraction. In this environment, models take input from and advise multiple operators while operators might interact with different models throughout a shift.

An integral element for implementing these systems is that operators are able to trust decisions made by their recommendation agents. This requires the system to be self-explaining in both adequate form and abstraction level. This involves both an explanation regarding the basis of the recommendation, e. g. what input parameters led to this output, as well as an assessment of the quality of the decision, e. g. what is the expected error in quality when executing the recommended parametrisation. In this paper, we posit that Learning Classifier Systems are well-suited to be used within the proposed supervised learning agent by reviewing different explainability techniques in light of this setting. We then highlight a variety of open research questions that need to be addressed to successfully apply LCSs (or other rule-based systems) in this context.

Learning Classifier Systems

Learning Classifier Systems (LCSs) are a family of rule-based learning systems (Urbanowicz and Moore, 2009). While LCSs are a diverse field, they share some common properties. In general, LCSs produce models consisting of a finite number of if-then rules (*classifiers*) where individual premises (*conditions*), and by extension the global model structure, are optimized using a—typically evolutionary—metaheuristic and the *conclusions* of the rules use a problem-dependent model. These classifiers or local models can then individually be ascribed a quality of their prediction within their respective subspace of the global model's input space. In our view, these commonalities are sufficient to motivate their application within a decision support system, however, we acknowledge that choosing the "right" LCS for an actual implementation needs to be done use-case specific as some LCSs will yield better results than others.

Explainability in LCSs

Explainbility of machine learning is usually differentiated into *transparent methods*, allowing interpretation of decisions and comprehension of the model from the structure itself, and *post-hoc methods*, utilising visualisation, model transformation into intrinsicly interpretable models and similar techniques on models that are not by themselves interpretable (Barredo Arrieta et al., 2020). As rule-based learning systems, LCSs generally fall into the domain of transparent models and are regarded as excellent for interpretability due to their relation to human behaviour. However, several factors can limit the degree to which humans can easily comprehend the model and follow its decision making process. Most notable are the number of classifiers and the formulation thereof. Conditions in complex feature spaces are harder to understand than those that operate directly on the data, e. g. higher level features aggregating multiple sensor readings versus the readings themselves. Additionally, conditions can be formulated using complex non-linear functions rather than readable decision boundaries (Bull and O'Hara, 2002). Conclusions that utilise complex black box models, such as neural networks (Lanzi and Loiacono, 2006), are also harder to understand than linear or constant models even if these local black box models are usually much smaller than a model of the same class that encompasses the complete problem space would need to be.

These issues can warrant design adjustments within the LCS or the application of post-hoc methods. The number of rules can be combated depending on the type of system considered: For Pittsburgh-style systems, this is usually achieved by promoting small individuals through adjustments of the fitness function (Bacardit and Garrell, 2007) whereas, in Michigan-style systems, rule subsumption and compaction methods are applied (Tan et al., 2013; Liu et al., 2019). An improved understanding of singular classifiers can be pursued by promoting simplicity during training through a suitable fitness function, by applying analysis typical for the respective models, e. g. feature importance estimations in neural networks, and with a variety of visualisation methods (Urbanowicz et al., 2012; Liu et al., 2019, 2021).

LCSs in Industrial decision support systems

Many different LCSs have been proposed over the years and while originally envisioned as a powerful reinforcement learner, they have been extended for all learning paradigms (Urbanowicz and Moore, 2009). We consider the application as a decision support system that proposes settings to

an operator and informs them of the reasoning behind this choice to be a supervised learning task. This can be solved with either online or offline learning as long as the model used to make recommendations provides a compacted version of itself for inference and subsequently serving explanations. The LCS learns from experiences including sensor readings, product information, used machine settings and resulting quality measures, all of which will be a mixture of real and categorical values. When tasked with assisting an operator, the SL agent uses sensor readings and product information to propose machine settings and predict the expected quality.

Besides the previously introduced explainability techniques, LCSs also easily allow us to provide operators with all examples from our training data that formed the local model (as we know which samples were matched by the classifier's condition). This can help further the trust that the model's predictions are actually based on existing expertise. Going beyond traditional explaining by example (Barredo Arrieta et al., 2020), each example that influenced this classifier's weights could theoretically be listed, whereas in black box models usually the entire sample influences every weight.

In Michigan-style LCSs, each individual classifier gets ascribed a quality measure (or multiple thereof in XCS(F)). This (or in case of multiple measures, at least one of them) represents the classifier's fitness and is used to guide the evolutionary process. Moreover, we can utilise these measures to provide our operator with additional information on how exact and therefore useful a recommendation is. Classifiers with a low prediction quality and thus a high expected error might provide poor machine settings while other classifiers in the model might actually provide very useful settings. This disparity in niches of the feature space can also allow insights into where new sampling should take place (Stein et al., 2017) and allows to differentiate the model further. Even if—viewed globally—the model is less than optimal, it can still be used within the SL agent and aid operators on tasks where it is well fit.

Open Research Questions

Following this theoretical examination of the applicability of LCSs as decision support systems for the parametrisation of industrial machinery in a complex socio-technical environment, we want to raise several open research questions we aim at answering in the coming years. Note that we broaden the scope from our operators that interact directly or indirectly with the machine to all stakeholders that have a vested interest in the operation of the shopfloor, both digital and analogue. Thus, this can also include regulatory bodies, safety officers, management, customers and others.

1. *To what extent does a stakeholder request explanations?* This can have numerous dimensions, such as depth, frequency or diversity of explanations. In this question we assume that stakeholders may seek explanations that go beyond regulatory requirements, although a potential answer may be that they are not interested in further/deeper explanations. This raises another aspect: How important is explainability deemed if prediction quality potentially suffers?

2. *What are the differences between types of stakeholders?* Tying directly into the previous question, we assume that the diverse stakeholders will answer this question differently. Someone that operates the machine directly might prefer examples of past experiences while quality assurance personnel might prefer visualisations or vice versa. Stakeholders may also hold different understandings of the machine itself, so explanations would need to accommodate specific levels of prior knowledge. Furthermore, diversity between individual operators might be substantial and warrant personalisation approaches.

3. *How many rules may the served model contain before being too large?* For the full model, smaller rule sets are easier to generate a general understanding on, while larger rule sets can provide a more diverse coverage of the input space and therefore more accurate and comprehensible predictions. In some cases, like explanations for specific decisions, the entirety of the rule set might not even be of interest and operators may prefer explanations to be limited to the rules whose conditions matched the situation.

4. *What form can conditions take before they are too complex to be understood?* Many rule representations have been proposed in the past and while ellipsoids or neural networks can provide improved results, cuboids might be easier to comprehend. This should also probe whether the exact condition is even considered relevant or if operators are content with knowing that it applies in this instance.

5. *How important are explanations of why the decision boundary of a classifier is placed a certain way?* In LCSs, the model structure (and decision boundary of each rule) is optimized using a metaheuristic to localise the classifiers in a way that they fit the data well. Within this question, we want to ascertain how important insights into this process are to operators.

6. *What form can conclusions take before they are too complex to be understood?* While linear models are widely regarded as easily comprehensible, more complex models might yield better results and typical explanations, such as feature importance analysis, can satisfy the operators' want for understanding the decision making process. This also translates to the usage of mixing models (where multiple classifiers are used to construct a prediction) and the comprehension thereof.

7. *What information do operators request about the training process?* Relating to question 5, this question aims towards the training in general and what steps are performed in the process rather than at an analysis of the utilised model.

Conclusion

In this paper, we introduced a complex socio-technical system within an industrial manufacturing setting where operators and supervised learning agents collaboratively adjust machine settings to optimize product quality. In these systems, operators can interact with a variety of heterogeneous machines and different agents throughout a single shift, while the agents also interact with different operators. Assisting the operators with recommendations from the agents decreases the necessity for experience and helps extract and conserve experience of senior operators that might otherwise be lost over time. We introduced Learning Classifier Systems (LCS) and reviewed why these rule-based systems are generally considered explainable. Building on that, we expanded on requirements for the design of an LCS within our agent and highlighted beneficial properties of LCSs for this application. This led to seven open research questions regarding the explainability and need thereof. Three of these questions are applicable to a variety of machine learning models, e. g. *To what extent does a stakeholder request explanations?*, and aim at analysing general wants and needs, while the other four questions are more specific for rule-based systems (LCSs, decision trees, etc.). Answers to these questions are likely very domain- and stakeholder-specific and would need to be answered for each manufacturing problem independently. Although we assume that general trends should be transferable, these questions can also serve as a template whenever applying rule-based learning systems to a new scenario where comprehensibility is essential. Consequently, we are confident that LCSs can introduce self-explaining into these socio-technical-systems while advancing industrial manufacturing practices.

References

Bacardit, J. and Garrell, J. M. (2007). Bloat control and generalization pressure using the minimum description length principle for a pittsburgh approach learning classifier system. In Kovacs, T., Llorà, X., Takadama, K., Lanzi, P. L., Stolzmann, W., and Wilson, S. W., editors, *Learning Classifier Systems*, pages 59–79, Berlin, Heidelberg. Springer Berlin Heidelberg.

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115.

Bull, L. and O'Hara, T. (2002). Accuracy-Based Neuro and Neuro-Fuzzy Classifier Systems. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, GECCO'02, page 905–911, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lanzi, P. and Loiacono, D. (2006). XCSF with Neural Prediction. In *2006 IEEE International Conference on Evolutionary Computation*, pages 2270–2276.

Liu, Y., Browne, W. N., and Xue, B. (2019). Absumption to complement subsumption in learning classifier systems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '19, page 410–418, New York, NY, USA. Association for Computing Machinery.

Liu, Y., Browne, W. N., and Xue, B. (2021). Visualizations for rule-based machine learning. *Natural Computing*.

Lughofer, E., Zavoianu, C., Pollak, R., Pratama, M., Meyer-Heye, P., Zörrer, H., Eitzinger, C., and Radauer, T. (2019). Autonomous Supervision and Optimization of Product Quality in a Multi-stage Manufacturing Process based on Self-adaptive Prediction Models. *Journal of Process Control*, 76:27–45.

Margraf, A., Stein, A., Engstler, L., Geinitz, S., and Hahner, J. (2017). An evolutionary learning approach to self-configuring image pipelines in the context of carbon fiber fault detection. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 147–154.

Permin, E., Bertelsmeier, F., Blum, M., Bützler, J., Haag, S., Kuz, S., Özdemir, D., Stemmler, S., Thombansen, U., Schmitt, R., Brecher, C., Schlick, C., Abel, D., Poprawe, R., Loosen, P., Schulz, W., and Schuh, G. (2016). Self-optimizing Production Systems. *Procedia CIRP*, 41:417–422.

Schoettler, G., Nair, A., Luo, J., Bahl, S., Ojea, J. A., Solowjow, E., and Levine, S. (2020). Deep Reinforcement Learning for Industrial Insertion Tasks with Visual Inputs and Natural Rewards. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5548–5555.

Stein, A., Maier, R., and Hähner, J. (2017). Toward curious learning classifier systems: Combining xcs with active learning concepts. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '17, page 1349–1356, New York, NY, USA. Association for Computing Machinery.

Tan, J., Moore, J., and Urbanowicz, R. (2013). Rapid Rule Compaction Strategies for Global Knowledge Discovery in a Supervised Learning Classifier System. In *ECAL 2013: The Twelfth European Conference on Artificial Life*, pages 110–117.

Urbanowicz, R. J., Granizo-Mackenzie, A., and Moore, J. H. (2012). An analysis pipeline with statistical and visualization-guided knowledge discovery for michigan-style learning classifier systems. *IEEE Computational Intelligence Magazine*, 7(4):35–45.

Urbanowicz, R. J. and Moore, J. H. (2009). Learning classifier systems: A complete introduction, review, and roadmap. *Journal of Artificial Evolution and Applications*, 2009.

Zhang, Y., Qian, C., Lv, J., and Liu, Y. (2017). Agent and cyber-physical system based self-organizing and self-adaptive intelligent shopfloor. *IEEE Transactions on Industrial Informatics*, 13(2):737–747.