Towards a Pittsburgh-style LCS for learning manufacturing machinery parametrizations

Michael Heider, David Pätzel, Jörg Hähner

Angaben zur Veröffentlichung / Publication details:

Heider, Michael, David Pätzel, and Jörg Hähner. 2020. "Towards a Pittsburgh-style LCS for learning manufacturing machinery parametrizations." In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion (GECCO '20), Cancún, Mexico, July 2020*, edited by Carlos Artemio Coello Coello, 127–28. New York, NY: ACM. https://doi.org/10.1145/3377929.3389963.

Towards a Pittsburgh-style LCS for Learning Manufacturing Machinery Parametrizations

Michael Heider*
University of Augsburg, Germany
michael.heider@informatik.
uni-augsburg.de

David Pätzel*
University of Augsburg, Germany
david.paetzel@informatik.
uni-augsburg.de

Jörg Hähner
University of Augsburg, Germany
joerg.haehner@informatik.
uni-augsburg.de

ABSTRACT

We present a first evaluation of a new accuracy-based Pittsburgh-style learning classifier system (LCS) for supervised learning of multi-dimensional continuous decision problems: The SupRB-1 (**Sup**ervised **R**ule-**B**ased) learning system. Designed primarily for finding parametrizations for industrial machinery, SupRB-1 learns an approximation of a continuous quality function from examples (consisting of situations, choices and associated qualities—all continous, the first two possibly multi-dimensional) and is then able to make an optimal choice as well as predict the quality of a choice in a given situation. This paper shows and discusses preliminary results of SupRB-1's performance on an additive manufacturing problem.

CCS CONCEPTS

• **Computing methodologies** → **Rule learning**; **Supervised learning**;

KEYWORDS

Learning Classifier Systems, Evolutionary Machine Learning, Manufacturing, Configuration, Rule-based Learning

ACM Reference Format:
Michael Heider, David Pätzel, and Jörg Hähner. 2020. Towards a Pittsburgh-style LCS for Learning Manufacturing Machinery Parametrizations. In *Genetic and Evolutionary Computation Conference Companion (GECCO '20 Companion), July 8–12, 2020, Cancún, Mexico.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3377929.3389963

1 MOTIVATION

Parametrization of industrial machinery is often determined by human operators who obtained most of their expertise through year-long experimental exploration based on prior knowledge about the system or process at play. Transferring that knowledge to other operators with as little loss as possible (e. g. to new colleagues whenever experienced operators retire) is a challenge: Humans' ability of exactly attributing parametrization choices to situations and communicating that knowledge tends to be rather restricted—which leads to new operators being forced to at least in part repeat said exploration to learn for themselves. While machine learning (ML) can help with this, in order to be met with the required acceptance and trust, a system to support this process needs to be able to transparently incorporate expert knowledge and have a human-understandable model representation. Parts of an operator's knowledge can be seen as a collection of mappings from parametrizations for the machine and variables beyond their influence to an expected process quality resulting from them—abstractly speaking, a collection of if-then rules with outcomes subject to noise. *Learning classifier systems* (LCSs) whose models are collections of human-readable *if-then rules* are ML techniques that are a natural fit for this problem [1].

This paper presents a preliminary evaluation of the SupRB-1 learning system, a new accuracy-based Pittsburgh-style LCS for supervised learning on continuous multi-dimensional decision problems such as the one of parametrization of industrial machinery. A global model which consists of multiple localized simplistic paraboloid (i. e. linear regression) models is evolved by a genetic algorithm that takes both prediction quality (accuracy) and model complexity (number of parameters to fit) into account when determining solution fitness by utilizing the *Bayesian Information Criterion* (BIC).

2 EVALUATION ON THE AM-GAUSS FUNCTION

We preliminarily investigate SupRB-1's performance on an abstract model of an FDM-based additive manufacturing (AM) process. The process itself consists of material (usually thermoplastic polymers) being melted and then extruded to gradually construct a part. At that, part quality is reliant on both outside situational influences, as well as a chosen parametrization for the given machine. We take the realistic, albeit simplified, assumption that the relationship between every such two factors is described by a Gaussian function meaning that there is an ideal combination for each pair (e. g. high ambient temperatures merit lower temperature parameters—but not too low).

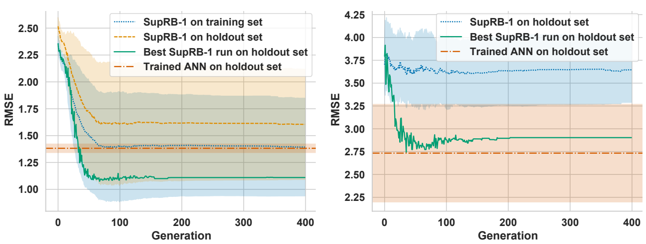
The FDM-based AM process we consider contains five continuous (a simplification by itself) situation dimensions: material, printer, room temperature, humidity and the kind of part to produce (written as $x_1, \ldots, x_5$). These situations interact with six continuous parameters $a_1, \ldots, a_6$ that form the parametrization: extrusion temperature, print bed temperature, cooling fan speed, extruder movement speed, material retraction speed and retraction distance (the first four parameters are self-explanatory; the latter two control behaviour when moving while not extruding). This leads to the following overall Gaussian mixture model for the quality function

*These authors contributed equally to the paper.

which we call an *AM-Gauss function*:

$$q(y) = \sum_{\substack{j \in 1,\ldots,11, \\ k \in 1,\ldots,11, \\ k \neq j}} \exp\left(-\left(\begin{pmatrix} y_j \\ y_k \end{pmatrix} - s_{j,k}\right)^T P_{j,k} \left(\begin{pmatrix} y_j \\ y_k \end{pmatrix} - s_{j,k}\right)\right) \quad (1)$$

where $y = (y_1, \ldots, y_{11})^T = (x_1, \ldots, x_5, a_1, \ldots, a_6)^T$, each $P_{j,k}$ is a positive semi-definite matrix in $\mathbb{R}^{2\times2}$ with eigenvalues in $[0, 30]$ (ensures sensible scaling) which describes the relationship between the respective two parameters and $s_{j,k}$ is a vector in $[-1, 1]^2$ specifying the location of the summand's mode. Note that we did not include noise in this first version of the model; however, an evaluation on more realistic noisy environments is already planned.



(a) ***Quality predictions*** **on training and holdout data with ANN baseline.**

(b) ***Parametrization choices*** **on holdout data with ANN baseline.**

Figure 1: Root mean squared error (with standard deviation (SD)) of different metrics on SupRB-1's elitist's performance, averaged over 20 runs on a single AM-Gauss function.

We generated one such AM-Gauss function from random seed 1 by randomly generating the required $P_{j,k}$'s and s's. For the resulting function, we created a training set containing 2000 examples (1000 for training local models, 1000 for optimizing global model structure). SupRB-1 was run 20 times with consecutive seeds for 400 generations on that data, after each iteration reporting its goodness-of-fit on an extra holdout set of size 1000 generated from the function as well. Additionally, for the situations in the holdout set (i. e. the first five dimensions of the inputs), SupRB-1 was tasked to predict the optimal parametrization (i. e. the second part of the input that maximizes the quality function); the quality of that parametrization was then compared with the best quality achievable in that situation according which was obtained by maximizing the quality function accordingly. The results of the runs' populations' elitists (the respective classifier populations with the highest fitness) are shown in Figure 1. We also investigated the performance on 29 other AM-Gauss functions (seeds 2 through 30), albeit with only one run each, achieving comparable results.

We compare SupRB-1's results with those achieved by a two-layer fully connected *artificial neural network* (ANN) evaluated on identical data. The ANN's exact architecture of 512 and 8 hidden cells while using ReLu activation functions twice was determined from performing a simple automated architecture optimization in terms of error during validation; model complexity was not factored into the architecture optimization strategy. We show the average of the performances of 20 such networks (again, using consecutive seeds for initialisation) after training on the holdout datasets as baselines.

Figure 1a shows that SupRB-1's quality predictions' RMSE on holdout data improves rapidly over the first 70 generations and then seems to converge at around 1.67 which falls short of the ANN baseline. In contrast to the average, the best run not only converges slightly faster, but also achieves much better results at an error of about 1.1 surpassing the baseline. Besides, some other runs also produced comparable results while about as many runs performed poorly and far from reaching the baseline. The most likely source of this is premature convergence to local optima. Some approaches to create and nurture a more healthy and diverse population of classifier populations and thereby hopefully decrease early convergence are already under investigation and will be reported on soon.

For the primary goal of being able to predict a good parametrization using the learned model the results are not as convincing. Examining the RMSE of the parametrization choices on holdout data (Figure 1b), it can be seen, that the average results only improve slightly over time and miss the baseline by far. However, the best run comes close to the baseline and is able to beat several ANNs performance-wise in the process. It is important to note here that, for SupRB-1, good results in prediction quality often seem to correlate with good results on choosing parametrizations, whereas many ANNs that performed good on the former had vastly worse results on the latter (a behaviour also indicated by the much larger standard deviation of the baseline in Figure 1b). We thus tentatively conjecture that, for this function and sample size, SupRB-1 generalizes more reliably to the task it did not explicitly learn (after all, it was only *trained* to *predict* quality and not optimize it) but this definitely has to be investigated in more detail.

The best individuals (i. e. classifier populations) of each run converge to 10 local models with a low standard deviation, the overall best performing individuals having 11 to 12 local models. As most runs evolved solutions of similar size but with substantial differences in errors we suspect that the models were ill placed at local optima, supporting the postulated premature convergence issue.

3 CONCLUSIONS

We presented a first evaluation of the SupRB-1 learning system, a general accuracy-based Pittsburgh-style LCS architecture for supervised learning of continuous multi-dimensional decision problems. We showed its applicability for finding a good machine parameterization choice for an abstract but highly complex model of an additive manufacturing process and compared the performance to a neural network. In terms of predicting the resulting quality for a given choice SupRB-1 achieved comparable and in some cases better results, while for the problem of choosing an optimal parametrization it was on average outperformed by the neural network. However, we already identified several ways to improve the system aside from solving the premature convergence issue, for example, by using non-parabolic local models, which we plan to pursue alongside the obligatory more in-depth general investigation.

REFERENCES

[1] Ryan J. Urbanowicz and Jason H. Moore. 2009. Learning Classifier Systems: A Complete Introduction, Review, and Roadmap. *Journal of Artificial Evolution and Applications* 2009 (2009).