Machine Learning and Configurable Systems: A Gentle Introduction

Hugo Martin, Juliana Alves Pereira, Mathieu Acher, Paul Temple

Machine Learning and Configurable Systems: A Gentle Introduction

Hugo Martin, Juliana Alves Pereira, Mathieu Acher
Univ Rennes, IRISA, Inria, CNRS, France

Paul Temple
PReCISE, NaDi
University of Namur, Belgium

ABSTRACT

The goal of this tutorial is to give an introduction to how machine learning can be used to support activities related to the engineering of configurable systems and software product lines. To the best of our knowledge, this is the first practical tutorial in this trending field. The tutorial is based on a systematic literature review and includes practical tasks (specialization, performance prediction) on real-world systems (VaryLaTeX, x264).

CCS CONCEPTS

• **Software and its engineering → Software product lines**;

KEYWORDS

Software Product Lines, Machine Learning, Configurable Systems

Configurable software systems and software product lines allow stakeholders to derive product variants that meet their specific functional and non-functional requirements. A straightforward way to find a suitable configuration is to measure the target (non-functional) property of each corresponding variant and then select any configuration that meets specified requirements. Unfortunately, enumerating and measuring all configurations is usually unfeasible due to the combinatorial explosion of possible variants and the cost of measurements. Machine learning techniques have gained momentum to predict the properties of configurable systems out of a sample – without measuring all variants. In this tutorial, we rely on the pattern *"sampling, measuring, learning, validation"* emerged in the software engineering and machine learning literature [3]. The usual process is to sample some configurations, execute and measure *e.g.*, their execution time, and learn out of measurements.

The practical tasks will be conducted on two case studies: VaryLaTeX [1] and x264 [2]. The VaryLaTeX case is used to illustrate how learning techniques can be used to *specialize* configurable systems through the automatic mining of constraints among options. We also use the video encoder *x264* to show how we can predict performance properties of unmeasured configurations [2, 3].

The interactive tutorial is a half-day event structured as follows:

Part 1 (theoretical, interactive) – Motivation, overview. Starting with the Linux kernel, attendees will understand how impossible it is to enumerate all possible product variants and the need to use alternative approaches, namely statistical machine learning. Based on our recent systematic literature survey [3], we will introduce a catalog of *"sampling, measurement, learning, validation"* techniques spanning different use-cases: performance prediction, configuration optimization, software understanding or specialization (constraint mining). Finally, we give concrete examples to illustrate the underlying challenges (e.g., resources cost vs error cost).

Part 2 (practical) – Learning constraints. In this session, we exercise how configurable systems can be *specialized* thanks to learning techniques used to automatically mine constraints among options. We use an intuitive example: a learning system, called VaryLaTeX [1]. VaryLaTeX is a solution based on variability, constraint programming, and machine learning techniques for documents written in LaTeX to meet constraints and deliver on time. This part aims to illustrate the end-to-end process (sampling, measuring, learning, validation) but also how to read and interpret decision trees which are used to specialize configurable systems.

Part 3 (practical) – Performance prediction. We show how we can predict performance properties (e.g., execution time) of unmeasured configurations of x264, a highly configurable video encoder considered in numerous works (see e.g., [2, 3]). With this part, attendees will understand how to frame a performance prediction problem as a regression problem and how machine learning algorithms can be applied.

Part 4 (theoretical) – Conclusion. We will recap all lessons learned, and point out limitations and open challenges that need attention in future work (e.g., how to select a significant sample of configurations? what is an ideal sample?).

The material (including slides, data, procedures) is available online: https://github.com/VaryVary/ML-configurable-SPLCTutorial/

Acknowledgements. This research was partially funded by the ANR-17-CE25-0010-01 VaryVary project and VeriLearn project grant EOS No. 30992574.

REFERENCES

[1] Mathieu Acher, Paul Temple, Jean-Marc Jézéquel, José A. Galindo, Jabier Martinez, and Tewfik Ziadi. 2018. VaryLATEX: Learning Paper Variants That Meet Constraints. In *VAMOS*. 83–88. DOI: http://dx.doi.org/10.1145/3168365.3168372

[2] Jianmei Guo, Krzysztof Czarnecki, Sven Apel, Norbert Siegmund, and Andrzej Wasowski. 2013. Variability-aware performance prediction: A statistical learning approach. In *ASE*.

[3] Juliana Alves Pereira, Hugo Martin, Mathieu Acher, Jean-Marc Jézéquel, Goetz Botterweck, and Anthony Ventresque. 2019. Learning Software Configuration Spaces: A Systematic Literature Review. (2019). arXiv:arXiv:1906.03018